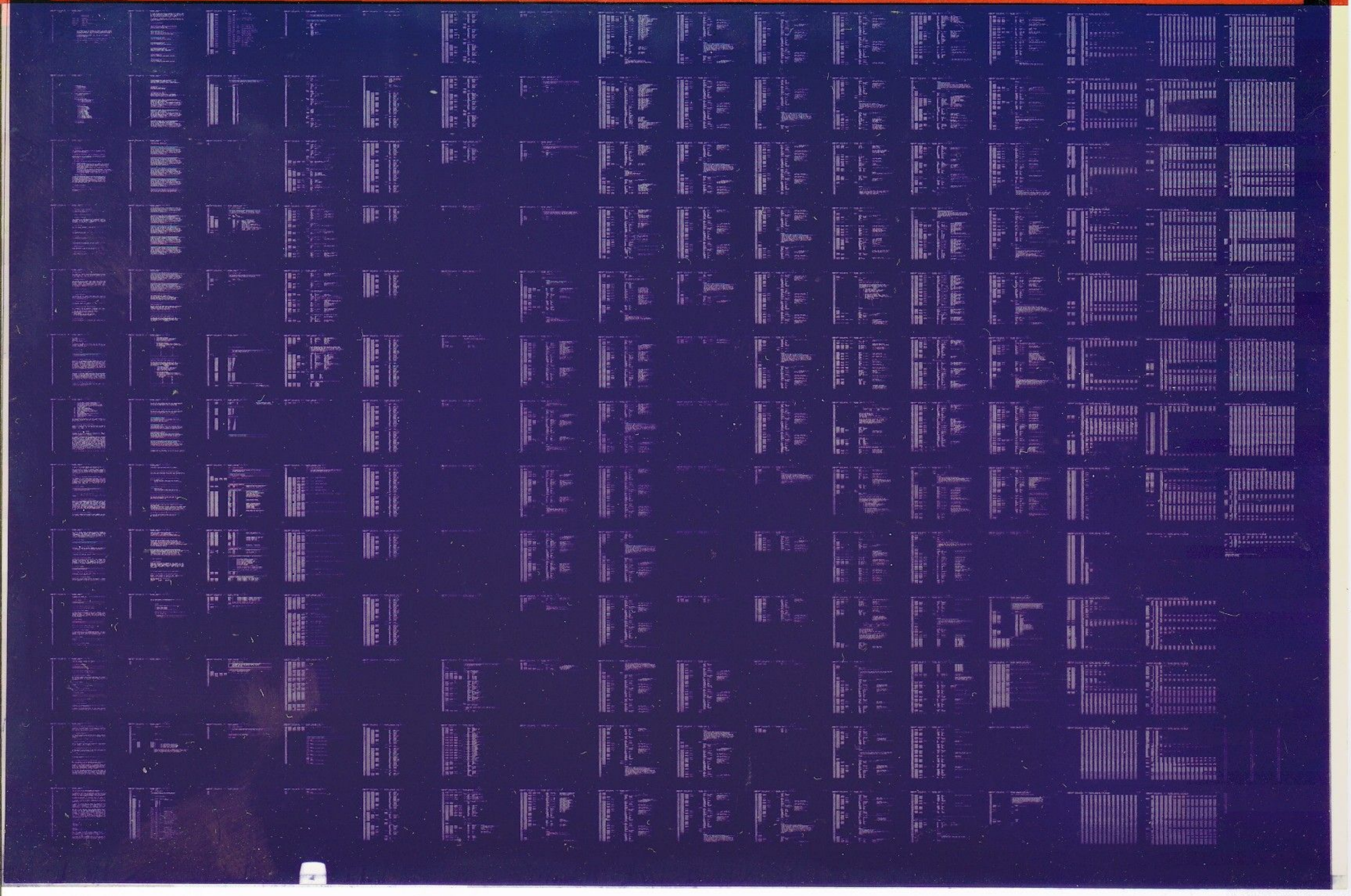
DMP-11,
DMR11,M8207

M8207 STATIC DIAG
CZDMQE0

AH-E229E-MC
FICHE 1 OF 1

MAY 1983

MADE IN USA



```
                        .REM @

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

                        PRODUCT CODE:   AC-E228E-MC

                        PRODUCT NAME:   CZDMQE0 M8207 STATIC DIAG #2

                        PRODUCT DATE:   OCTOBER 1982

                        MAINTAINER:     DIAGNOSTICS MERRIMACK

                        AUTHOR:         ED BADGER
```

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 PROGRAM ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST OUT THE M8200, M8204, OR 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 REVISION HISTORY

THIS DIAGNOSTIC WAS REVISED FOR THE FOLLOWING REASONS

REV A - ORIGINAL RELEASE
REV B - CORRECT AN ERROR TYPEOUT IN TEST 26, CORRECT TIME DEPENDENT
        CODE IN TEST 20, CORRECT MACHINE DEPENDENT CODE IN TEST 37
REV C - FIX NECESSARY FOR DMP MICROCODE CHANGE. MICROCODE CHANGE
        CAUSED TEST 43 TO FAIL
REV D - NEW MICROCODE CAUSED ERRONEOUS DATA TO REMAIN IN HIGH BYTE
        OF R5 IN TEST 40.
REV E - CHANGED TEST 26 AT THE REQUEST OF MANUFACTURING. THIS CHANGE
        WILL ALLOW TESTING OF BITS 4 AND 5 IN IBUS*<13> (PC BITS 12 AND 13).
        THE DIAGNOSTIC ORIGINALLY ONLY TESTED A 12 BIT PC. THIS DID NOT
        EFFECT THE DMR; HOWEVER THE DMP-YA DOES USE A 14 BIT PC.
        THE CHANGE WAS PROPOSED BY MARTY DIMUZIO OF AUGUSTA MANUF.

1.3 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 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
                        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-D-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
                        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:LOE

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>/PASS:<PASS-CNT>/FLAGS:
   <FLAG-LIST>/UNITS:<UNIT-LIST>
**************************************************************
```

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 REM' DER 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 (SEE 0) 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: (0) 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).


                        7.0  TEST DESCRIPTIONS


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


                        ********************* 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.
                        ************************************************


                        ********************* TEST 3 *******************
                        *IOP CRAM WRITE/READ TEST
                        *FLOAT A 1 THROUGH EACH CRAM LOCATION
                        ************************************************


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


                        ********************* TEST 5 *******************
                        *IOP CRAM DUAL ADDRESSING TEST
                        *WRITE EACH ADDRESS INTO ITSELF, READ EACH
                        *ADDRESS TO VERIFY CORRECT ADDRESSING
                        ************************************************


                        ********************* TEST 6 *******************
                        *IOP MAIN MEMORY TEST
                        *FLOAT A 1 THROUGH ALL MAIN MEMORY LOCATIONS
                        ************************************************


                        ********************* TEST 7 *******************
                        *IOP MAIN MEMORY TEST
                        *FLOAT A 0 THROUGH ALL MAIN MEMORY LOCATIONS
                        ************************************************
```

```
761
762
763                     ********** TEST 8 **********
764                     *IOP MAIN MEMORY DUAL ADDRESSING TEST
765                     *LOAD EACH MEMORY LOCATION WITH ITS OWN ADDRESS
766                     *READ BACK EACH LOCATION TO VERIFY CORRECT ADDRESSING
767                     ************************************************
768
769
770                     ********** TEST 9 **********
771                     *IOP MAR TEST
772                     *PERFORM DUAL ADDRESSING TEST
773                     *USING MAR AUTO-INC FEATURE
774                     ************************************************
775
776
777                     ********** TEST 10 **********
778                     *IOP (CRAM) ODT BITS TEST
779                     *LOAD MAR WITH A 0 INC MAR UNTIL IT OVERFLOWS
780                     *VERIFY THAT IBUS* 10 BITS IS SET ONLY WHEN MAR BIT 8 IS A ONE
781                     *AND THAT IBUS* 10 BIT6 IS SET ON MAR OVERFLOW
782                     ************************************************
783
784
785                     ********** TEST 11 **********
786                     *CRAM TEST OF JUMP(I) NEVER MICRO-PROCESSOR INSTRUCTION.
787                     *PERFORM THE JUMP INSTRUCTION
788                     *VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
789                     *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
790                     *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
791                     *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
792                     *THE CRAM PC IS CORRECT.  IF THE CRAM PC IN NOT RIGHT,
793                     *THEN PORT4 CONTAINS A 37
794                     ************************************************
795
796
797                     ********** TEST 12 **********
798                     *CRAM TEST OF JUMP(I) ALWAYS MICRO-PROCESSOR INSTRUCTION.
799                     *PERFORM THE JUMP INSTRUCTION
800                     *VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
801                     *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
802                     *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
803                     *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
804                     *THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
805                     *THEN PORT4 WILL CONTAIN A 37
806                     ************************************************
807
808
809                     ********** TEST 13 **********
810                     *CRAM TEST OF JUMP(I) ON C BIT SET MICRO-PROCESSOR INSTRUCTION.
811                     *SET THE C BIT, PERFORM THE JUMP INSTRUCTION.
812                     *VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
813                     *IN THE LOCATION IT IS AT.  THIS INSTRUCITON LOADS THE
814                     *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
815                     *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
816                     *THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
```

```
                        *THEN PORT4 WILL CONTAIN A 37.
                        ************************************************


                        ******************** TEST 14 ***********************
                        *CRAM TEST OF JUMP(I) ON Z BIT SET MICRO-PROCESSOR INSTRUCTION.
                        *SET 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 15 ***********************
                        *CRAM TEST OF JUMP(I) ON BR0 SET MICRO-PROCESSOR INSTRUCTION.
                        *SET 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.  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 THE PORT4 WILL CONTAIN A 37
                        ************************************************


                        ******************** TEST 16 ***********************
                        *CRAM TEST OF JUMP(I) ON BR1 SET MICRO-PROCESSOR INSTRUCTION.
                        *SET 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
                        *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 17 ***********************
                        *CRAM TEST OF JUMP(I) ON BR4 SET MICRO-PROCESSOR INSTRUCTION.
                        *SET THE BR4 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 18 ***********************
                        *CRAM TEST OF JUMP(I) ON BR7 SET MICRO-PROCESSOR INSTRUCTION.
                        *SET THE BR7 BIT, PERFORM THE JUMP INSTRUCTION
                        *VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
```

```
873                             *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
874                             *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
875                             *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
876                             *THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
877                             *THEN PORT4 WILL CONTAIN A 37
878                             ****************************************************
879
880
881                             ******************** TEST 19 ************************
882                             *CRAM TEST OF JUMP(I) ON C BIT CLEAR MICRO-PROCESSOR INSTRUCTION.
883                             *SET THE C BIT, PERFORM THE JUMP INSTRUCTION.
884                             *VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
885                             *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
886                             *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
887                             *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
888                             *THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
889                             *THEN PORT4 WILL CONTAIN A 37
890                             ****************************************************
891
892
893                             ******************** TEST 20 ************************
894                             *CRAM TEST OF JUMP(I) ON Z BIT CLEAR MICRO-PROCESSOR INSTRUCTION.
895                             *CLEAR THE Z BIT, PERFORM THE JUMP INSTRUCTION.
896                             *VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
897                             *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
898                             *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
899                             *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
900                             *THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
901                             *THEN PORT4 WILL CONTAIN A 37
902                             ****************************************************
903
904
905                             ******************** TEST 21 ************************
906                             *CRAM TEST OF JUMP(I) ON BR0 CLEAR MICRO-PROCESSOR INSTRUCTION.
907                             *CLEAR THE BR0 BIT, PERFORM THE JUMP INSTRUCTION.
908                             *VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
909                             *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
910                             *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  A THIS POINT
911                             *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
912                             *THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
913                             *THEN PORT4 WILL CONTAIN A 37
914                             ****************************************************
915
916
917                             ******************** TEST 22 ************************
918                             *CRAM TEST OF JUMP(I) ON BR1 CLEAR MICRO-PROCESSOR INSTRUCTION.
919                             *CLEAR THE BR1 BIT, PERFORM THE JUMP INSTRUCTION.
920                             *VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
921                             *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
922                             *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
923                             *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
924                             *THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
925                             *THEN PORT4 WILL CONTAIN A 37
926                             ****************************************************
927
928
```

```
                        ********** 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.
                        * REV. E - CHANGE TO TEST BITS 4 & 5 (PC BITS 12 & 13)
                        *
                        ************************************************


                        ********** TEST 27 **********
                        *
                        *JUMP TEST, JUMP ALWAYS, JUMP CHANGE FIELD
                        *
                        *IN THIS TEST, WE WILL CHECK THE ABILITY OF THE
                        *MICRO-PROCESSOR TO JUMP (BRANCH AND ALWAYS INSTRUCTION)
                        *TO LOCATIONS, FIELDS FROM OTHER LOCATIONS FIELDS.
                        *WE ALREADY KNOW THAT THE BRANCH INSTR WORKS FROM
                        *OTHER TEST.             PROCEDURE:
```

```
                        *      1.  START ADDR 0, FIELD 0
                        *      2.  **CALCULATE NEW ADDRS, FIELD VIA INC,
                        *      3.  CAUSE JUMP (BRANCH) TO NEW ADDRESS
                        *      4.  READ PC FROM IBUS*12 AND IBUS*13
                        *      5.  REPEAT STEP 2-4 256.TIMES
                        *
                        *      TO CALCULATE NEW ADDRESS:
                        *      1.  INC LOW BYTE OF ADDRESS FOR PC ADDRESS 0-7
                        *      2.  INC LOW BYTE OF N ADDRESS FOR PC ADDRESS 8-11
                        *          BITS REPRESENTED AS BITS 0-3.  WHEN 0-3 OVERFLOWS,
                        *          RESTARTS AT ZERO.
                        *                NET RESULT IS JUMPS FROM:
                        *          FIELD,PAGE                    LOC
                        *                0                       0
                        *                1                       1
                        *                2                       2
                        *                3                       3
                        *               10                       7
                        *               11                      11
                        *               :TO                     :
                        *               17                     377
                        ************************************************

                        ****************** TEST 28 ********************
                        *
                        *JUMP TEST, JUMP ALWAYS, JUMP CHANGE FIELD
                        *
                        *IN THIS TEST, WE WILL CHECK THE ABILITY OF THE
                        *MICRO-PROCESSOR TO JUMP (BRANCH AND ALWAYS INSTRUCTION)
                        *TO LOCATIONS, FIELDS FROM OTHER LOCATIONS FIELDS.
                        *WE ALREADY KNOW THAT THE BRANCH INSTR WORKS FROM
                        *OTHER TESTS.              PROCEDURE:
                        *      1.  START ADDR 0, FIELD 0
                        *      2.  **CALCULATE NEW ADDRS, FIELD VIA DEC,
                        *      3.  CAUSE JUMP (BRANCH) TO NEW ADDRESS
                        *      4.  READ PC FROM IBUS*12 AND IBUS*13
                        *      5.  REPEAT STEP 2-4 256.TIMES
                        *
                        *      TO CALCULATE NEW ADDRESS:
                        *      1.  DEC LOW BYTE OF ADDRESS FOR PC ADDRESS 0-7
                        *      2.  DEC LOW BYTE OF N ADDRESS FOR PC ADDRESS 8-11
                        *          BITS REPRESENTED AS BITS 0-3.  WHEN 0-3 OVERFLOWS,
                        *          RESTARTS AT ZERO
                        *              NET RESULT IS JUMPS FROM:
                        *            FIELD,PAGE                LOC:
                        *                  0                   0
                        *                 17                   377
                        *                 16                   376
                        *                 15                   375
                        *                 :TO                  :
                        *                 00                   000
                        ************************************************

                        ****************** TEST 29 ********************
```

```
1041                                    *
1042                                    *IN THIS TEST WE'LL VERIFY THAT THE Z BIT CAN BE READ FROM
1043                                    *IBUS*<13>. WE ALREADY KNOW THAT THE Z BIT WORKS PROPERLY,
1044                                    *ALL WE WANT TO KNOW HERE IS THAT IT CAN BE READ.
1045                                    ********************************************************
1046
1047
1048
1049                                    ****************** TEST 30 ***********************
1050                                    *
1051                                    *IN THIS TEST WE'LL VERIFY THAT THE C BIT CAN BE READ FROM
1052                                    *IBUS*<13>. WE ALREADY KNOW THAT THE C BIT WORKS PROPERLY
1053                                    *ALL WE WANT TO KNOW HERE IS THAT IT BE READ.
1054                                    *
1055                                    ********************************************************
1056
1057
1058                                    ****************** TEST 31 ***********************
1059                                    *TEST OF PROGRAM CLOCK BIT
1060                                    *DO A MASTER CLEAR, VERIFY THAT PROGRAM CLOCK IS SET
1061                                    *WRITE PROGRAM CLOCK BIT TO A ONE, VERIFY THAT IT CLEARS,
1062                                    *AND THEN SETS SOME TIME LATER
1063                                    ********************************************************
1064
1065
1066                                    ****************** TEST 32 ***********************
1067                                    *FORCE POWER FAIL TEST
1068                                    *SET FORCE POWER FAIL BIT VERIFY THAT PROCESSOR TRAPS TO 24
1069                                    *GOING DOWN AND COMING UP.  VERIFY ALSO THAT BUS INIT WAS
1070                                    *BLOCKED FROM GETTING TO THE M8200,4,7 DURING THE POWER FAIL
1071                                    ********************************************************
1072
1073
1074                                    ****************** TEST 33 ***********************
1075                                    *MICRO-PROCESSOR NOISE TEST
1076                                    *WRITE ALL ZERO'S THEN ALL ONE'S THEN A DATA PATTERN
1077                                    *TO THE IBUS* AND IBUS REGISTERS AND TO THE SP AND MAIN MEM
1078                                    *THEN GO BACK AND READ THE DATA PATTERNS TO VERIFY THAT
1079                                    *READING AND WRITING OF OTHER LOCATIONS AND REGISTERS
1080                                    *DID NOT CHANGE THE DATA.
1081                                    ********************************************************
1082
1083
1084                                    ****************** TEST 34 ***********************
1085                                    *THIS TEST IS DESIGNED TO MAKE SURE THAT A NODST INSTRUCTION
1086                                    *DOES NOT WRITE INTO PORT B OF THE MULTIPORT RAM.
1087                                    *TO DO THIS, WE'LL PUT A 125 INTO INDAT2, THEN WE'LL PUT A
1088                                    *125 INTO BOTH SP1 AND BR. LAST WE'LL DO A NODST BR, SUBOC, SP1
1089                                    *IF THERE IS A WRITE INTO PORTB, INADT2 WILL CONTAIN A 377
1090                                    ********************************************************
1091
1092
1093                                    ****************** TEST 35 ***********************
1094                                    *
1095                                    *EXTENDED CRAM TEST FOR M8206.  IN THIS TEST WE WILL LOAD DATA
1096                                    *THROUGHOUT THE CRAM (TEST DATA IS JUST 4K OF DIAG. CODE) AND
```

```
1097                                    *THEN READ IT BACK AND VERIFY THAT IT IS CORRECT
1098                                    ************************************************
1099
1100
1101                                    ****************** TEST 36 ************************
1102                                    *
1103                                    *THIS TEST LOADS MICRO-CODE INTO A M8206 MCPU THEN EXECUTES IT.
1104                                    *THE MICRO-CODE IS DESIGNED TO WRITE ALL ONES INTO THE SEL REGS.
1105                                    *
1106                                    ************************************************
1107
1108
1109                                    ****************** TEST 37 ************************
1110                                    *
1111                                    *NEGATIVE ADDRESS TEST.
1112                                    *       IN THIS TEST, WE'LL MAKE SURE THAT THE M8207
1113                                    *       DOES NOT RESPOND TO AN ADDRESS THAT ISN'T ASSIGNED
1114                                    *        TO IT
1115                                    *
1116                                    ************************************************
1117
1118
1119                                    ****************** TEST 38 ************************
1120                                    *
1121                                    *BYTE ADDRESSING TEST
1122                                    *       HERE, WE'RE GOING TO MAKE SURE THAT WE CAN
1123                                    *        WRITE INTO ONLY A HIGH OR LOW BYTE OF THE MCPU.
1124                                    *
1125                                    ************************************************
1126
1127
1128                                    ****************** TEST 39 ************************
1129                                    *
1130                                    *IN THIS TEST WE'RE GOING TO MAKE SURE THAT THE PC
1131                                    *REG COUNTS UP PROPERLY.  THE PC REG SHOULD INCREMENT
1132                                    *ONCE AFTER EACH INSTRUCTION.
1133                                    *
1134                                    ************************************************
1135
1136
1137                                    ****************** TEST 40 ************************
1138                                    *
1139                                    *IN THIS TEST WE'LL MAKE SURE THAT "BRANCH FIELD H" DOESN'T
1140                                    *GET STUCK HIGH.
1141                                    *FIRST WE'LL CLEAR THE PC HIGH REG. THEN WE'LL DO A BRANCH INSTR
1142                                    *WITH BAB BITS 11+12 SET.  IF PCR  BITS 8+9 SET THEN WE'LL KNOW
1143                                    *WE WERE SUCCESSFUL IF PCR BITS 8+9 FAIL TO SET, WE,LL KNOW
1144                                    *THAT THE MAX SELECTED THE WRONG INPUT TO BE CLOCKED INTO THE PCR.
1145                                    ************************************************
1146
1147
1148                                    ****************** TEST 41 ************************
1149                                    *
1150                                    *IN THIS TEST WE'RE GOING TO MAKE SURE THAT ONLY SPO
1151                                    *IS SELECTED FOR SOURCE WHEN THE DESTINATION
1152                                    *IS THE OUTBUS
```

```
1153                    *FIRST WE'LL WRITE EACH SP ADDRS INTO ITSELF THEN WE'LL
1154                    *MOV SP TO OBUS4.  THAT SHOULD SELECT
1155                    *SP ADDRESS 0.  IF ANY OTHER DATA SHOWS UP, WE'LL
1156                    *BLAME IT ON THE SELECTION OF A DIFFERENT SCRATCH PAD.
1157                    ****************************************************
1158
1159
1160                    ****************** TEST 42 ****************************
1161                    *
1162                    *IN THIS TEST WE ARE GOING TO MAKE SURE THAT THE
1163                    *SIGNAL "MOV INST H" (AND ITS ASSOC. TRIBS) DOESN'T GET
1164                    *STUCK HIGH.  IN ORDER TO DO THIS WE'LL CLEAR THE PC HIGH REG
1165                    *PUT KNOWN DATA IN THE BREG AND SP1 THEN WE'LL BRANCH
1166                    *WITH CROM BITS 0-3 SET AS WELL AS CROM BIT 9 WITH CROM BITS 8 AND 11 CLEAR.
1167                    *IF "MOV INST H" GETS STUCK HIGH, THE PC REG HIGH WILL GET LOADED
1168                    *WITH THE CONTENTS OF THE ALU
1169                    ****************************************************
1170
1171
1172                    ****************** TEST 43 ****************************
1173                    *TEST THAT MASTER CLEAR, CLEARS BITS IN THE NPR CONTROL REGISTER AND
1174                    *MICROPROCESSOR MISCELLANEOUS REGISTER-FIRST WE'LL SET THE
1175                    *PRIORITY UP SO THAT WHEN WE SET THE BUS REQUEST BIT THAT IT WON'T BUG US
1176                    *THEN WE'LL SET ALL THE BITS IN BOTH REGS EXCEPT THE
1177                    *NPR REQUEST. WE'LL LOOK TO SEE THAT ALL GOT SET, NEXT
1178                    *WE'LL DO A MASTER CLEAR AND BE SURE THAT THEY ALL CLEAR.
1179                    ****************************************************
1180
1181
1182
1183                    8.0 ERROR INFORMATION
1184
1185
1186                    8.1 ERROR REPORTING
1187
1188                    ERRORS ARE REPORTED BY THE PROGRAM AS  THEY  OCCUR  (IF  NOT
1189                    INHIBITED). THE REPORT CONFORMS TO THE DIAGNOSTIC SUPERVISOR
1190                    ERROR REPORT FORMAT, AND CONSISTS OF A  DESCRIPTION  OF  THE
1191                    ERROR,  THE  TEST  NUMBER,  SUBTEST  NUMBER, PC OF THE ERROR
1192                    CALL,  DEVICE  ADDRESS,  AND  BASIC  AND  EXTENDED  ERROR
1193                    INFORMATION.
1194
1195
1196
1197                    THE FOLLOWING EXAMPLES PROVIDE TYPICAL ERROR REPORTS:
1198
1199
1200                    CZDMQ DVC FTL ERR 00045 TST 027 SUB 000 PC:022572
1201
1202                    MASTER  CLEAR  FAILED  TO  CLEAR  PC  REG,  CONTENTS=000624
1203                    CZDMQ DVC FTL ERR 00015 TST 042 SUB 000 PC:027234
1204
1205                    UNIT=00, FAILING UNIT ADDRESS=160170
1206                    JUMP TEST ERROR
1207                    FROM ADDR     TO ADDR      BAD ADDR
1208                    000402        000000       000114
```

```
1209
1210
1211
1212                    FOR ALL OTHER ERRORS, THE REPORT MAY BE MORE  EXTENSIVE  AND
1213                    REQUIRE ADDITIONAL DATA TO BE REPORTED.
1214
1215
1216
1217                    9.0     HISTORY
1218
1219                       -    MODIFIED AUGUST 1980 FOR THE FOLLOWING REASONS:
1220
1221                            1) CANCELL DEPO CZDMQA1
1222                            2) CANCELL DEPO CZDMQA2
1223                            3) DETECT BAD TIMING ON INTERNAL CLOCK.
1224
1225
1226                       -    MODIFIED JULY 1981 TO FIX TEST 43 MAR BITS IN IBUS* 10.
1227
1228                       -    MODIFIED JANUARY 1982
1229
1230                            1) ERRONEOUS DATA WAS NOT CLEARED IN HIGH BYTE OF REGISTER 5
1231                               IN TEST 40.
1232                               FIX:  CHANGE BIC  #374,R5   TO BIC #177774,R5.
1233
1234                    @
```

```
1235
1236
1237
1238
1239
```

```
1240                                    .TITLE CZDMQE0 M8207 STATIC DIAG #2
1241            002000                          .=2000
1242
1243
1244
1245
1246
1247
1248                                    .MCALL  SVC
1249  002000                            SVC                             ; INITIALIZE SUPERVISOR MACROS
1250
1251
1252
1253
1254
1255  002000                            BGNMOD  CZDMQ
1256
1257
1258            000000                  $LSTIN= 0
1259            000000                  $LSTTAG= 0
1260            000000                  SVCINS= 0       ; LIST INSTRUCTIONS, SHIFTED RIGHT
1261            000000                  SVCTST= 0       ; LIST TEST TAGS, SHIFTED RIGHT
1262            000000                  SVCSUB= 0       ; LIST SUBTEST TAGS, SHIFTED RIGHT
1263            000000                  SVCGBL= 0       ; LIST GLOBAL TAGS, SHIFTED RIGHT
1264            000000                  SVCTAG= 0       ; LIST OTHER TAGS, SHIFTED RIGHT
1265
1266                                    ;       CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
1267                                    ;       TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS.  CHANGE THE
1268                                    ;       SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS.  YOU MAY
1269                                    ;       CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
1270
1271
```

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

                                        POINTER BGNAU,BGNDU


                                        HEADER  CZDMQ,D,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  104                             .ASCII  /D/
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  027344                          .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  030144                          .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
```

```
1328  002052
1329  002052  000000          L$EF::                  ;DIAG. EVENT FLAGS
                              .WORD   0
1330  002054  000000          .WORD   0
1331  002056                  L$SPC::
1332  002056  000000          .WORD   0
1333  002060                  L$DEVP::                ; POINTER TO DEVICE TYPE LIST
1334  002060  002730          .WORD   L$DVTYP
1335  002062                  L$REPP::                ;PTR. TO REPORT CODE
1336  002062  000000          .WORD   0
1337  002064                  L$EXP4::
1338  002064  000000          .WORD   0
1339  002066                  L$EXP5::
1340  002066  000000          .WORD   0
1341  002070                  L$AUT::                 ;PTR. TO ADD UNIT CODE
1342  002070  012144          .WORD   L$AU
1343  002072                  L$DUT::                 ;PTR. TO DROP UNIT CODE
1344  002072  012140          .WORD   L$DU
1345  002074                  L$LUN::                 ;LUN FOR EXERCISERS TO FILL
1346  002074  000000          .WORD   0
1347  002076                  L$DESP::                ;POINTER TO DIAG. DESCRIPTION
1348  002076  002312          .WORD   L$DESC
1349  002100                  L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
1350  002100  104035          EMT     E$LOAD
1351  002102                  L$ETP::                 ;POINTER TO ERRTBL
1352  002102  000000          .WORD   0
1353  002104                  L$ICP::                 ;PTR. TO INIT CODE
1354  002104  011340          .WORD   L$INIT
1355  002106                  L$CCP::                 ;PTR. TO CLEAN-UP CODE
1356  002106  012134          .WORD   L$CLEAN
1357  002110                  L$ACP::                 ;PTR. TO AUTO CODE
1358  002110  012042          .WORD   L$AUTO
1359  002112                  L$PRT::                 ;PTR. TO PROTECT TABLE
1360  002112  002122          .WORD   L$PROT
1361  002114                  L$TEST::                ;TEST NUMBER
1362  002114  000000          .WORD   0
1363  002116                  L$DLY::                 ;DELAY COUNT
1364  002116  000000          .WORD   0
1365  002120                  L$HIME::                ;PTR. TO HIGH MEM
1366  002120  000000          .WORD   0
1367
1368
1369  002122                  BGNPROT
1370  002122                  L$PROT::
1371  002122  177777          .WORD   -1
1372  002124  177777          .WORD   -1
1373  002126  177777          .WORD   -1
1374  002130                  ENDPROT
1375
```

```
1376                                    .SBTTL  DISPATCH TABLE
1377
1378                                    ;////////////////////////////////////////////////////////////////////////////
1379                                    ;/ THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
1380                                    ;/ IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
1381                                    ;////////////////////////////////////////////////////////////////////////////
1382
1383  002130                                    DISPATCH 43
1384  002130  000053                           .WORD   43
1385  002132                            L$DISPATCH::
1386  002132  012146                           .WORD   T1
1387  002134  012256                           .WORD   T2
1388  002136  012422                           .WORD   T3
1389  002140  012552                           .WORD   T4
1390  002142  012712                           .WORD   T5
1391  002144  013154                           .WORD   T6
1392  002146  013356                           .WORD   T7
1393  002150  013570                           .WORD   T8
1394  002152  014112                           .WORD   T9
1395  002154  014470                           .WORD   T10
1396  002156  014736                           .WORD   T11
1397  002160  015202                           .WORD   T12
1398  002162  015432                           .WORD   T13
1399  002164  015676                           .WORD   T14
1400  002166  016142                           .WORD   T15
1401  002170  016406                           .WORD   T16
1402  002172  016652                           .WORD   T17
1403  002174  017116                           .WORD   T18
1404  002176  017362                           .WORD   T19
1405  002200  017642                           .WORD   T20
1406  002202  020122                           .WORD   T21
1407  002204  020376                           .WORD   T22
1408  002206  020654                           .WORD   T23
1409  002210  021130                           .WORD   T24
1410  002212  021404                           .WORD   T25
1411  002214  021576                           .WORD   T26
1412  002216  021744                           .WORD   T27
1413  002220  022322                           .WORD   T28
1414  002222  022560                           .WORD   T29
1415  002224  022774                           .WORD   T30
1416  002226  023210                           .WORD   T31
1417  002230  023452                           .WORD   T32
1418  002232  024044                           .WORD   T33
1419  002234  025116                           .WORD   T34
1420  002236  025216                           .WORD   T35
1421  002240  025364                           .WORD   T36
1422  002242  026042                           .WORD   T37
1423  002244  026202                           .WORD   T38
1424  002246  026310                           .WORD   T39
1425  002250  026446                           .WORD   T40
1426  002252  026544                           .WORD   T41
1427  002254  026644                           .WORD   T42
1428  002256  026770                           .WORD   T43
1429
1430
1431
```

```
1432
1433
1434
```

```
                        .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
                                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

                                ENDHW
002310                  L10001:
```

```
                                .SBTTL  SOFTWARE P-TABLE

                                ;////////////////////////////////////////////////////////////////////////
                                ;/ THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
                                ;/ PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
                                ;////////////////////////////////////////////////////////////////////////

002310                                  BGNSW   SFPTBL
002310  000000                          .WORD   L10002-L$SW/2
002312                          L$SW::
002312                          SFPTBL::


002312                                  ENDSW
002312                          L10002:
```

```
                                .SBTTL  GLOBAL EQUATES SECTION

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

002312                                  EQUALS
                                ;
                                ; BIT DIFINITIONS
                                ;
100000                          BIT15== 100000
040000                          BIT14== 40000
020000                          BIT13== 20000
010000                          BIT12== 10000
004000                          BIT11== 4000
002000                          BIT10== 2000
001000                          BIT09== 1000
000400                          BIT08== 400
000200                          BIT07== 200
000100                          BIT06== 100
000040                          BIT05== 40
000020                          BIT04== 20
000010                          BIT03== 10
000004                          BIT02== 4
000002                          BIT01== 2
000001                          BIT00== 1
                                ;
001000                          BIT9==  BIT09
000400                          BIT8==  BIT08
000200                          BIT7==  BIT07
000100                          BIT6==  BIT06
000040                          BIT5==  BIT05
000020                          BIT4==  BIT04
000010                          BIT3==  BIT03
000004                          BIT2==  BIT02
000002                          BIT1==  BIT01
000001                          BIT0==  BIT00
                                ;
                                ; EVENT FLAG DEFINITIONS
                                ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                                ;
000040                          EF.START==      32.                             ; START COMMAND WAS ISSUED
000037                          EF.RESTART==    31.                             ; RESTART COMMAND WAS ISSUED
```

```
1541    000036          EF.CONTINUE==   30.                      ; CONTINUE COMMAND WAS ISSUED
1542    000035          EF.NEW==        29.                      ; A NEW PASS HAS BEEN STARTED
1543    000034          EF.PWR==        28.                      ; A POWER-FAIL/POWER-UP OCCURRED
1544                    ;
1545                    ;
1546                    ; PRIORITY LEVEL DEFINITIONS
1547                    ;
1548    000340          PRI07== 340
1549    000300          PRI06== 300
1550    000240          PRI05== 240
1551    000200          PRI04== 200
1552    000140          PRI03== 140
1553    000100          PRI02== 100
1554    000040          PRI01== 40
1555    000000          PRI00== 0
1556                    ;
1557                    ;OPERATOR FLAG BITS
1558                    ;
1559    000004          EVL==       4
1560    000010          LOT==      10
1561    000020          ADR==      20
1562    000040          IDU==      40
1563    000100          ISR==     100
1564    000200          UAM==     200
1565    000400          BOE==     400
1566    001000          PNT==    1000
1567    002000          PRI==    2000
1568    004000          IXE==    4000
1569    010000          IBE==   10000
1570    020000          IER==   20000
1571    040000          LOE==   40000
1572    100000          HOE==  100000
1573
1574
1575
1576
1577                    ;;*****************************************************************
1578                    ;* PROGRAM EVENT FLAG DEFINITIONS
1579                    ;;*****************************************************************
1580
1581
1582
```

```
1583                                    .SBTTL  GLOBAL DATA SECTION
1584
1585                                    ;////////////////////////////////////////////////////////////////////////////
1586                                    ;/      THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1587                                    ;/      IN MORE THAN ONE TEST.
1588                                    ;////////////////////////////////////////////////////////////////////////////
1589
1590                                    ;;**************************************************************************
1591                                    ;* STORAGE FOR DEVICE REGISTERS
1592                                    ;;**************************************************************************
1593  002312                            DESCRIPT        <M8207 DIAG. #2 OF 2>
1594  002312                            L$DESC::
1595  002312  034115  030062  020067            .ASCIZ  /M8207 DIAG. #2 OF 2/
1596  002320  044504  043501  020056
1597  002326  031043  047440  020106
1598  002334  000062
1599
1600                                            .EVEN
1601
1602                                    ;**************************************************************************
1603                                    ;* PROGRAM CONTROL PARAMETERS
1604                                    ;**************************************************************************
1604  002336  000000                    NEXT:   .WORD   0               ;ADDRESS OF NEXT TEST TO BE EXECUTED
1605  002340  000000                    LOCK:   .WORD   0               ;ADDRESS FOR LOCK CURRENT DATA
1606
1607                                    ;**************************************************************************
1608                                    ;* MISCELLANEOUS STORAGE
1609                                    ;**************************************************************************
1610  002342  000000                    LOGDEV: .WORD   0               ;LOGICAL DEVICE NUMBER
1611  002344  000000                    PSTACK: .WORD   0               ;BASE LEVEL PROGRAM STACK POINTER
1612  002346  000000                    SUBRPC: .WORD   0               ;PC OF SUBR CALL FOR ERROR REPORTS
1613  002350  000000                    ERRFLG: .WORD   0               ;SUBROUTINE ERROR FLAG
1614  002352  000000                    RETADR: .WORD   0               ;SUBR ERROR RETURN ADDRESS
1615  002354  000000                    STRTSW: .WORD   0               ;SWITCHES AT START OF PROGRAM
1616  002356  000000                    STAT:   .WORD   0               ;KM STATUS WORD STORAGE
1617  002360  000000                    CLKX:   .WORD   0               ;
1618  002362  000000                    MASKX:  .WORD   0               ;
1619  002364  000000                    SAVSP:  .WORD   0               ;STACK POINTER STORAGE
1620  002366  000000                    SAVPC:  .WORD   0               ;PROGRAM COUNTER STORAGE
1621  002370  000000                    ZERO:   .WORD   0               ;
1622  002372  000001                    ONE:    .WORD   1               ;
1623  002374  000000                    MEMLIM: .WORD   0               ;HIGHEST LOCATION FOR NPR'S
1624  002376  000001                    KMACTV: .BLKW   1               ;M8200,4,7 SELECTED ACTIVE
1625  002400  000001                    KMNUM:  .BLKW   1               ;OCTAL NUMBER OF M8200,4,7'S
1626  002402  000001                    SAVACT: .BLKW   1               ;ORIGINAL ACTIVE DEVICES
1627  002404  000001                    SAVNUM: .BLKW   1               ;WORKABLE NUMBER
1628  002406  000000                    FLAG:   .WORD   0               ;SCRATCH STORAGE
1629  002410  000000                    RUN:    .WORD   0               ;POINTER TO RUNNING DEVICES
1630  002412  000000                    FADR:   .WORD   0
1631  002414  000000                    MTYPE:  .WORD   0               ;M82XX NUMBER FOR TYPE OF MICO-CPU
1632  002416  000000                    $REG5:  .WORD   0               ;STORAGE USED FOR ERROR MSG DATA
1633  002420  000000                    $REG4:  .WORD   0
1634  002422  000000                    $REG3:  .WORD   0
1635  002424  000000                    $REG2:  .WORD   0
1636  002426  000000                    $REG1:  .WORD   0
1637  002430  000000                    $REG0:  .WORD   0
1638  002432  000000                    TYPE:   .WORD   0               ;=0 FOR DMP,=1 FOR M8206
```

```
1639  002434  000000                  MRO:    .WORD   0               ;MEMLOC USED INSTEAD OF R0.
1640  002436  003777                  MEMSZ:  .WORD   3777            ;INDICATES MEMORIE SIZE, LAST ADDR.
1641  002440  000000                  TEMP:   .WORD   0
1642  002442  000000                  $TEMP0: .WORD   0
1643  002444  000000                  $TMP0:  .WORD   0
1644  002446  000000                  $GDADR: .WORD   0               ;CONTAINS ADDRESS OF 'GOOD' DATA
1645  002450  000000                  $BDADR: .WORD   0               ;CONTAINS ADDRESS OF 'BAD' DATA
1646  002452  000000                  $GDDAT: .WORD   0               ;CONTAINS 'GOOD' DATA
1647  002454  000000                  $BDDAT: .WORD   0               ;CONTAINS 'BAD' DATA
1648  002456  000000                          .WORD   0               ;RESERVED--NOT TO BE USED
1649  002460  000000                          .WORD   0
1650  002462  000000                  FTIME:  .WORD   0
1651  002464  000000                  SAVE4:  .WORD   0
1652  002466  000000                  SAVE6:  .WORD   0
1653  002470  000000                  RUNB:   .WORD   0               ;0= RUN OFF, 1= RUN SW ON
1654  002472  000000                  RUNINH: .WORD   0               ;0=RUN SW OFF, 1=RUN SW ON
1655
1656                                  ;*******************************************************************
1657                                  ;* PROGRAM CONTROL FLAGS
1658                                  ;*******************************************************************
1659  002474     000                  INIFLG: .BYTE   0               ;PROGRAM INITIALIZING FLAG
1660        002476                            .EVEN
1661  002476     000                  LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
1662  002477     000                  QV.FLG: .BYTE   0               ;QUICK VERIFY FLAG
1663                                          .EVEN
1664
1665                                  ;*******************************************************************
1666                                  ;* DEFINITION OF M8200,4,7 STATUS WORDS - STAT1,STAT2,STAT3
1667                                  ;*
1668                                  ;*
1669                                  ;*      STAT1 - BITS 00-08 IS M8200,4,7 VECTOR ADDRESS
1670                                  ;*              BIT15=1 LINE UNIT IS AN M8203
1671                                  ;*              BIT14=0 NO TEST CONNECTOR(S) USED
1672                                  ;*              BIT14=1 H-XXX TEST CONNECTOR WILL BE USED
1673                                  ;*              BIT13=0 LINE UNIT IS AN M8201
1674                                  ;*              BIT13=1 LINE UNIT IS AN M8202
1675                                  ;*              BIT12=1 NO LINE UNIT
1676                                  ;*              BITS 09-11 IS M8200,4,7 PRIORITY LEVEL
1677                                  ;*
1678                                  ;*      STAT2 - LOW BYTE IS SWITCH PACK #1 (DDCMP LINE NUMBER)
1679                                  ;*              HIGH BYTE IS SWITCH PACK #2 (BM873 BOOT ADDRESS)
1680                                  ;*
1681                                  ;*      STAT3 - BIT0=1 DO FREE RUNNING TESTS ON M8200,4,7
1682                                  ;*
1683                                  ;*******************************************************************
1684  002500  000000                  STAT1:  .WORD   0
1685  002502  000000                  STAT2:  .WORD   0
1686  002504  000000                  STAT3:  .WORD   0
1687
1688                                  ;*******************************************************************
1689                                  ;* POINTERS TO M8200,4,7 VECTORS AND REGISTERS
1690                                  ;*******************************************************************
1691  002506  000000                  KMRVEC: 0       ;POINTER TO M8200,4,7 RCV INTRPT VECTOR
1692  002510  000000                  KMRLVL: 0       ;POINTER TO M8200,4,7 RCV INTRPT SERVICE PS
1693  002512  000000                  KMTVEC: 0       ;POINTER TO M8200,4,7 TX INTRPT VECTOR
1694  002514  000000                  KMTLVL: 0       ;POINTER TO M8200,4,7 TX INTRPT SERVICE PS
```

```
1695  002516  000000                  KMCSR:  0       ;POINTER TO M8200.4.7 CONTROL STATUS REGISTER
1696  002520  000000                  KMCSRH: 0       ;POINTER TO M8200.4.7 CONTROL STATUS REGISTER HIGH BYTE
1697  002522  000000                  KMCTL:  0       ;POINTER TO M8200.4.7 CONTROL OUT REGISTER
1698  002524  000000                  KMP04:  0       ;POINTER TO M8200.4.7 PORT REGISTER - SEL4
1699  002526  000000                  KMP06:  0       ;POINTER TO M8200.4.7 PORT REGISTER - SEL6
1700
1701                                  ;;****  PRIMARY REG ADRS STORAGE FOR THIS UNIT  ******
1702                                  ;THESE LOCATIONS WILL BE LOADED FOR THE CURRENT UNIT, IN INIT CODE
1703  002530                          REGADR:
1704
1705                                  ;;****  STACK USED FOR SUBROUTINE LINKAGE  *****
1706  002530  000100                          .BLKW   100
1707  002730                          SSTACK:
1708
1709
1710
1711
1712
1713
1714
```

```
                                        .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
                                        ;
```

```
1744                            .SBTTL  GLOBAL SUBROUTINES
1745
1746                            ;////////////////////////////////////////////////////////////////////////////////
1747                            ;/      THE GLOBAL SUBROUTINES ARE CALLED BY MORE THAN ONE TEST
1748                            ;////////////////////////////////////////////////////////////////////////////////
1749
1750
1751
1752                            ;-------------------------------------------------------------
1753                            ; MACRO'S NEEDED TO CALL SUBROUTINES
1754                            ;-------------------------------------------------------------
1755
1756                            .MACRO  POPSP2
1757                                    22626
1758                            .ENDM
```

```
1759
1760
1761
1762
1763
```

```
1764
1765                    ;////////////////////////////////////////////////////////////////////////////
1766                    ;/      THE GLOBAL SUBROUTINES ARE CALLED BY MORE THAN ONE TEST
1767                    ;////////////////////////////////////////////////////////////////////////////
1768
1769
1770
1771                    ;-------------------------------------------------------------
1772                    ; MACRO'S NEEDED TO CALL SUBROUTINES
1773                    ;-------------------------------------------------------------
1774
1775                    .MACRO  K4ONLY  ?N2
1776                            CMP     MEMSZ,#2000
1777                            BNE     N2
1778                            EXIT    TST
1779                            .ENDM
1780                    .MACRO  ED$CALL XY
1781                            .LIST
1782                            ;************** TEST 'XY' **************
1783                            .NLIST
1784                            .ENDM
1785                            .MACRO  BADHEAD
1786                            .RADIX  10
1787                            ED$CALL \T$TESTNUM+1
1788                            .RADIX  8
1789                            .ENDM
```

```
1790                            .MACRO  MYINT
1791                                    .LIST
1792                                    MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
1793                                    .NLIST
1794                            .ENDM
1795
1796                            .MACRO  MACEX   ?N2
1797                                    .LIST
1798                                    ;DO NOT DO TEST IF M8200
1799                                    .NLIST
1800                                    TST     TYPE
1801                                    BNE     N2
1802                                    EXIT    TST
1803                            N2:
1804                            .ENDM   MACEX
1805                            .MACRO  MACEX2  ?N2
1806                                    .LIST
1807                                    ;DO NOT DO TEST IF M8200
1808                                    .NLIST
1809                                    CMP     WTYPE,#0
1810                                    BNE     N2
1811                                    EXIT TST
1812                            N2:
1813                            .ENDM   MACEX2
1814                            .MACRO  K4ONLY  ?N2
1815                                    .LIST
1816                                    ;DO NOT DO TEST IF M8200, OR M8204
1817                                    .NLIST
1818                                    CMP     MEMSZ,#2000
1819                                    BNE     N2
1820                                    EXIT    TST
1821                            N2:
1822                                                            ;NOTE THIS TEST IS ONLY DESIGNED FOR 4K MODULE.
1823                            .ENDM
1824
1825                            .MACRO  CLRMAR
1826                                    ROMCLK
1827                                    004000
1828                            .ENDM   CLRMAR
1829                            .MACRO  ROMCLK
1830                            .LIST
1831                                    JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
1832                            .NLIST
1833                            .ENDM
1834
1835                            .MACRO  SROMCLK
1836                                    .LIST
1837                                    JSR     R5,.SROMCLK
1838                                    .NLIST
1839                            .ENDM
1840                            .MACRO  SKIP06  NNN
1841                                    .LIST
1842                                    ;GOTO 'NNN' IF M8206
1843                                    .NLIST
1844                                    CMP     WTYPE,#6                ;SEE IF M8206
1845                                    BEQ     NNN
```

```
                                        .ENDM
                                        .MACRO  SKIP07  NNN
                                                .LIST
                                                ;GOTO 'NNN' IF M8207
                                                .NLIST
                                                CMP     WTYPE,#7                ;SEE IF M8200,4,7
                                                BEQ     NNN
                                        .ENDM
                                        .MACRO  SKIP04  NNN
                                                .LIST
                                                ;GOTO 'NNN' IF M8204
                                                .NLIST
                                                CMP     WTYPE,#4                ;SEE IF M8204
                                                BEQ     NNN
                                        .ENDM
                                        .MACRO  MSTCLR
                                        JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
                                        .ENDM

002756                                  .MSTCLR:
002756  112777  000100  177534                 MOVB    #BIT6,@KMCSRH   ;SET INST.
002764  142777  000300  177526                 BICB    #BIT6!BIT7,@KMCSRH
002772  000205                                 RTS     R5

002774  000024                  PATCH:  .BLKW   20.             ;PATCH AREA.


003044                          ENDBUG:
                                ;       UNSAFE TO PATCH ANY OTHER AREA.
003044                          .ROMCLK:
003044  000240                          NOP
003046  000240                          NOP
003050  152777  000002  177442  .REGT:  BISB    #BIT1,@KMCSRH
003056  012577  177444                  MOV     (R5)+,@KMP06
003062  152777  000003  177430          BISB    #BIT1!BIT0,@KMCSRH
003070  142777  000007  177422          BICB    #BIT2!BIT1!BIT0,@KMCSRH
003076  000205                          RTS     R5

003100                          .SROMCLK:
003100  000240                          NOP
003102  022737  000006  002414          CMP     #6,WTYPE
003110  001357                          BNE     .REGT
003112  152777  000002  177400          BISB    #BIT1,@KMCSRH
003120  012577  177402                  MOV     (R5)+,@KMP06
003124  000240                          NOP
003126  000240                          NOP
003130  142777  000007  177362          BICB    #7,@KMCSRH
003136                          1$:
003136  152777  000001  177354          BISB    #BIT0,@KMCSRH           ;STEP INSTR.
003144  142777  000007  177346          BICB    #BIT2!BIT1!BIT0,@KMCSRH
003152  000240                          NOP
003154  000240                          NOP
003156  152777  000002  177334          BISB    #2,@KMCSRH
003164                          2$:
003164  000205                          RTS     R5
```

```
1902  003166                          CLRALL:
1903
1904  003166                                  ;CLEARS C & Z BITS AND BR
1905  003166  004537  003044                  ROMCLK
1906  003172  000400                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
1907  003174                          400                     ;0 TO BR
1908  003174  004537  003044                  ROMCLK
1909  003200  063220                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
1910  003202                          63220                   ;SP(0) TO BR
1911  003202  004537  003044                  ROMCLK
1912  003206  060400                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
1913  003210                          60400                   ;BR,SP(0) + BR
1914  003210  004537  003100                  SROMCLK
1915  003214  000000                  JSR     R5,..SROMCLK
1916  003216  000207                  0
1917                                  RTS     PC
1918  003220                          SETBR0:
1919
1920  003220                                  ;SETS BR0 BIT
1921  003220  004537  003044                  ROMCLK
1922  003224  000401                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
1923  003226  000207                  401                     ;1 TO BR
1924                                  RTS     PC
1925  003230                          SETBR1:
1926
1927                                  ;THIS SUBROUTINE SETS BR1 BIT
1928  003230                          ROMCLK                  ;NEXT WORD IS INSTRUCTION
1929  003230  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
1930  003234  000402                  000402                  ;BR_002
1931  003236  000207                  RTS     PC
1932
1933  003240                          SETBR4:
1934
1935                                  ;THIS SUBROUTINE SETS BR4 BIT
1936  003240                          ROMCLK                  ;NEXT WORD IS INSTRUCTION
1937  003240  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
1938  003244  000420                  420
1939  003246  000207                  RTS     PC
1940
1941  003250                          SETBR7:
1942
1943                                  ;THIS SUBROUTINE SETS BR7 BIT
1944  003250                          ROMCLK                  ;NEXT WORD IS INSTRUCTION
1945  003250  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
1946  003254  000600                  600
1947  003256  000207                  RTS     PC
1948
1949  003260                          SETC:
1950
1951                                  ;THIS SUBROUTINE SETS THE C BIT
1952  003260                          ROMCLK                  ;NEXT WORD IS INSTRUCTION
1953  003260  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
1954  003264  000777                  000777                  ;BR 377
1955  003266                          ROMCLK                  ;NEXT WORD IS INSTRUCTION
1956  003266  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
1957  003272  063220                  063220                  ;SP(0)_BR
```

```
1958  003274
1959  003274  004537  003044                ROMCLK                          ;NEXT WORD IS INSTRUCTION
1960  003300  060400                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
1961  003302                                060400                          ;BR SP(0)+BR
1962  003302  004537  003100                SROMCLK                         ;NOW WE MUST CLOCK THE BITS INTO IBUS <13>
1963  003306  000000                        JSR     R5,..SROMCLK
1964  003310  000207                        0
1965                                        RTS     PC
1966  003312                          SETZ:
1967                                        ;THIS SUBROUTINE SETS THE Z BIT
1968
1969  003312                                ROMCLK                          ;NEXT WORD IS INSTRUCTION
1970  003312  004537  003044                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
1971  003316  000777                        000777                          ;BR 377
1972  003320                                SROMCLK                         ;NOW CLOCK THE BITS INTO IBUS<13>
1973  003320  004537  003100                JSR     R5,..SROMCLK
1974  003324  000777                        0777
1975  003326  000207                        RTS     PC
1976
1977  003330                          RAMDAT:
1978                                        ;THIS SUBROUTINE LOADS R4 WITH THE LOWEST
1979                                        ;8 BITS OF THE CRAM PC.
1980
1981  003330  005004                        CLR     R4
1982  003332  017605  000000                MOV     @(SP),R5                ;GOOD DATA
1983  003336  062716  000002                ADD     #2,(SP)                 ;ADJUST STACK
1984  003342                                SKIP06  1$      ;IF M8206,WE'LL GET PC A DIFFERENT WAY.
1985                                        ;GOTO 1$ IF M8206
1986  003352                                SKIP07  1$      ;IF M8200,4,7 WE'LL GET PC A DIFFERENT WAY.
1987                                        ;GOTO 1$ IF M8207
1988  003362  005011                        CLR     (R1)                    ;CLEAR BIT10
1989  003364  052711  000400                BIS     #BIT8,(R1)              ;CLOCK INSTRUCTION IN CRAM THAT
1990                                                                        ;JUMPED TO, IT LOADS BR WITH IT
1991  003370  005011                        CLR     (R1)                    ;CLR BIT8
1992  003372                                ROMCLK                          ;NEXT WORD IS INSTRUCTION
1993  003372  004537  003044                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
1994  003376  061225                        061225                          ;MOV BR TO PORT 5
1995  003400  116104  000005                MOVB    5(R1),R4                ;PUT "FOUND" IN R4
1996  003404  000207                        RTS     PC                      ;RETURN
1997
1998  003406                          1$:   ROMCLK                          ;READ PC LOW REG DIRECTLY.
1999  003406  004537  003044                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
2000  003412  121244                        121244                          ;IBUS*<12> TO PORT 4
2001  003414  116104  000004                MOVB    4(R1),R4                ;PUT INTO R4
2002  003420  000207                        RTS PC                          ;EXIT
2003
2004  003422                          WROM:
2005                                        ;THIS SUBROUTINE WRITES THE ROMMAP INTO THE CRAM
2006
2007                                ;       BIT     #BIT15,STAT1            ;BE SURE M8200,4,7 HAS CRAM
2008                                ;       BEQ     2$                      ;SKIP IF NO CRAM
2009  003422                                SKIP07  2$
2010                                        ;GOTO 2$ IF M8207
2011  003432  005000                        CLR     R0                      ;R0=CRAM ADDRESS
2012  003434  012702  012146                MOV     #ROMMAP,R2              ;R2 POINTS TO ROMMAP
2013  003440  012711  002000          1$:   MOV     #BIT10,(R1)             ;SET ROM0
```

```
2014  003444  010061  000004                MOV     R0,4(R1)        ;LOAD CRAM ADDRESS
2015  003450  012261  000006                MOV     (R2)+,6(R1)     ;LOAD WORD TO BE WRITTEN
2016  003454  052711  020000                BIS     #BIT13,(R1)     ;WRITE IT!
2017  003460  005200                        INC     R0              ;NEXT ADDRESS
2018  003462  023700  002436                CMP     MEMSZ,R0        ;DONE YET?
2019  003466  001364                        BNE     1$              ;BR IF NO
2020  003470  005011                        CLR     (R1)            ;CLEAR SEL0
2021  003472  000207              2$:       RTS     PC              ;RETURN
2022
2023  003474                    MEMSET:
2024                                        ;THIS SUBROUTINE LOADS CRAM WITH SPECIAL INSTRUCTIONS
2025                                        ;FOR THE CRAM JUMP TEST. ALL CRAM LOCATIONS ARE LOADED
2026                                        ;WITH INSTRUCTIONS THAT MOVE A 37 TO THE BR, EXCEPT THE
2027                                        ;FOLLOWING CRAM ADDRESSES: 0,1,4,7,525,1777. THESE LOCATIONS
2028                                        ;CONTAIN INSTRUCTIONS WHICH LOAD THE BR WITH THE LOWEST
2029                                        ;8 BITS OF THAT CRAM ADDRESS.
2030
2031  003474                                SKIP07  3$      ;IF M8200,4,7 CAN'T WRITE CRAM!
2032                                        ;GOTO 3$ IF M8207
2033  003504  005000                        CLR     R0              ;R0 = CRAM ADDRESS
2034  003506  012711  002000          1$:   MOV     #BIT10,(R1)     ;SET ROMO
2035  003512  010061  000004                MOV     R0,4(R1)        ;LOAD CRAM ADDRESS
2036  003516  012761  000437  000006        MOV     #437,6(R1)      ;LOAD INSTRUCTION
2037  003524  052711  020000                BIS     #BIT13,(R1)     ;WRITE INSTRUCTION IN CRAM
2038  003530  005200                        INC     R0              ;NEXT ADDRESS
2039  003532  023700  002436                CMP     MEMSZ,R0        ;DONE YET?
2040  003536  001363                        BNE     1$              ;BR IF NO
2041  003540  005000                        CLR     R0              ;INDEX REGISTER
2042  003542  012711  002000          2$:   MOV     #BIT10,(R1)     ;SET ROMO
2043  003546  016061  003602  000004        MOV     CRAMA(R0),4(R1) ;LOAD CRAM ADDRESS IN SEL4
2044  003554  016061  003616  000006        MOV     INSTU(R0),6(R1) ;LOAD INSTRUCTION TO BE WRITTEN
2045  003562  052711  020000                BIS     #BIT13,(R1)     ;WRITE CRAM!
2046  003566  005720                        TST     (R0)+           ;NEXT
2047  003570  022700  000014                CMP     #14,R0          ;DONE YET?
2048  003574  001362                        BNE     2$              ;BR IF NO
2049  003576  005011                        CLR     (R1)            ;CLEAR ALL BITS
2050  003600  000207              3$:       RTS     PC              ;RETURN
2051
2052  003602  000000  000001  000004  CRAMA:  .WORD   0,1,4,7,1777,525
2053  003610  000007  001777  000525
2054
2055  003616  000400                  INSTU:  000400          ;BR_0
2056  003620  000401                          000401          ;BR_1
2057  003622  000404                          000404          ;BR_4
2058  003624  000407                          000407          ;BR_7
2059  003626  000777                          000777          ;BR_377
2060  003630  000525                          000525          ;BR_125
2061
2062
2063                                        ;ROUTINE TO SAVE GENERAL REGISTERS FOR ERROR ROUTINE.
2064                                        ; CALL = JSR    PC,SV05
2065  003632  010537  002416          SV05:  MOV     R5,$REG5
2066  003636  010437  002420                 MOV     R4,$REG4
2067  003642  010337  002422                 MOV     R3,$REG3
2068  003646  010237  002424                 MOV     R2,$REG2
2069  003652  010137  002426                 MOV     R1,$REG1
```

```
 2070  003656  013737  002434  002430          MOV     @R0,$REGO
 2071  003664  000207                          RTS     PC
 2072
 2073
```

```
                                .SBTTL  GLOBAL ERROR REPORT SECTION

                                ;////////////////////////////////////////////////////////////////
                                ;/      THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
                                ;/      THAT ARE USED IN MORE THAN ONE TEST.
                                ;////////////////////////////////////////////////////////////////

003666  047045  047445  022466  TFM1:   .ASCIZ  /%N%06%S4%06%S4%04%N/
003674  032123  047445  022466
003702  032123  047445  022464
003710  000116
003712  047045  047445  022463  TFM2:   .ASCIZ  /%N%03%S7%03%N/
003720  033523  047445  022463
003726  000116
003730  047045  047445  022463  TFM3:   .ASCIZ  /%N%03%S10%03%S4%04%N/
003736  030523  022460  031517
003744  051445  022464  032117
003752  047045     000
003755     045  022516  031517  TFM4:   .ASCIZ  /%N%03%S7%03%N/
003762  051445  022467  031517
003770  047045     000
003773     045  022516  033117  TFM5:   .ASCIZ  /%N%06%S5%06%S3%06%N/
004000  051445  022465  033117
004006  051445  022463  033117
004014  047045     000
004017     045  022516  051101  TFM36:  .ASCIZ  /%N%AREGISTER ADDRESS ERROR,ADDRESS = %06%A,UNIT = %02/
004024  043505  051511  042524
004032  020122  042101  051104
004040  051505  020123  051105
004046  047522  026122  042101
004054  051104  051505  020123
004062  020075  047445  022466
004070  026101  047125  052111
004076  036440  022440  031117
004104     000
004105     045  022516  020101  TFM41:  .ASCIZ  /%N%A CSR HIGH BYTE GOT WRITTEN INTO ON A LOW BYTE XFER/
004112  051503  020122  044510
004120  044107  041040  052131
004126  020105  047507  020124
004134  051127  052111  042524
004142  020116  047111  047524
004150  047440  020116  020101
004156  047514  020127  054502
004164  042524  054040  042506
004172  000122
004174  047045  040445  041440  TFM42:  .ASCIZ  /%N%A CSR LOW BYTE GOT WRITTEN INTO ON A HIGH BYTE XFER/
004202  051123  046040  053517
004210  041040  052131  020105
004216  047507  020124  051127
004224  052111  042524  020116
004232  047111  047524  047440
004240  020116  020101  044510
```

```
2130  004246  044107  041040  052131
2131  004254  020105  043130  051105
2132  004262     000
2133  004263     045  022516  047101  TFM40:  .ASCIZ  /%N%ANEG ADDR TEST DUAL ADDR ERROR-BAD ADDR = %06/
2134  004270  043505  040440  042104
2135  004276  020122  042524  052123
2136  004304  042040  040525  020114
2137  004312  042101  051104  042440
2138  004320  051122  051117  041055
2139  004326  042101  040440  042104
2140  004334  020122  020075  047445
2141  004342  000066
2142  004344  040445  051440  051103  TFM43:  .ASCIZ  /%A SCRATCH PAD %03%A DUAL ADDRESS ERROR WITH SP%02/
2143  004352  052101  044103  050040
2144  004360  042101  022440  035517
2145  004366  040445  042040  040525
2146  004374  020114  042101  051104
2147  004402  051505  020123  051105
2148  004410  047522  020122  044527
2149  004416  044124  051440  022520
2150  004424  031117     000
2151  004427     045  022524  052101  TFM44:  .ASCIZ  /%T%ATHE MAR REG, CONTENTS= %06/
2152  004434  042510  046440  051101
2153  004442  051040  043505  020054
2154  004450  047503  052116  047105
2155  004456  051524  020075  047445
2156  004464  000066
2157  004466  052045  040445  044124  TFM45:  .ASCIZ  /%T%ATHE PC REG, CONTENTS= %06/
2158  004474  020105  041520  051040
2159  004502  043505  020054  047503
2160  004510  052116  047105  051524
2161  004516  020075  047445  000066
2162  004524  047045  040445  047516  TFM45A: .ASCII  /%N%ANOTE: THIS ERROR MAY BE FALSELY GENERATED IF THE/
2163  004532  042524  020072  044124
2164  004540  051511  042440  051122
2165  004546  051117  046440  054501
2166  004554  041040  020105  040506
2167  004562  051514  046105  020131
2168  004570  042507  042516  040522
2169  004576  042524  020104  043111
2170  004604  052040  042510
2171  004610  047045  040445  052522          .ASCIZ  /%N%ARUN BIT (SW7 OF E28) IS ON/
2172  004616  020116  044502  020124
2173  004624  051450  033527  047440
2174  004632  020106  031105  024470
2175  004640  044440  020123  047117
2176  004646     000
2177  004647     045  047101  051120  TFM46:  .ASCIZ  "%ANPR/MISC REGS DATA FAILURE, GOOD =%06%A, BAD -%06"
2178  004654  046457  051511  020103
2179  004662  042522  051507  042040
2180  004670  052101  020101  040506
2181  004676  046111  051125  026105
2182  004704  043440  047517  020104
2183  004712  022475  033117  040445
2184  004720  020054  040502  020104
2185  004726  022475  033117     000
```

```
2186  004733     045  050101  020103  TFM47:  .ASCIZ  "%APC INCR. INCORRECT: S/B= %C6%A ; WAS = %06"
2187  004740  047111  051103  020056
2188  004746  047111  047503  051122
2189  004754  041505  035124  051440
2190  004762  041057  020075  047445
2191  004770  022466  020101  020073
2192  004776  040527  020123  020075
2193  005004  047445  000066
2194  005010  040515  052123  051105  TMMC:   .ASCIZ  /MASTER CLEAR FAILED TO CLEAR /
2195  005016  041440  042514  051101
2196  005024  043040  044501  042514
2197  005032  020104  047524  041440
2198  005040  042514  051101  000040
2199  005046  047045  052045  047045  FM1:    .ASCIZ  /%N%T%N/
2200  005054     000
2201
2202
2203
2204  005055     000                  EM0:    .ASCIZ  //
2205  005056  051103  046501  042040  EM1:    .ASCIZ  /CRAM DATA ERROR/
2206  005064  052101  020101  051105
2207  005072  047522  000122
2208  005076  051103  046501  042040  EM2:    .ASCIZ  /CRAM DUAL ADDRESSING ERROR/
2209  005104  040525  020114  042101
2210  005112  051104  051505  044523
2211  005120  043516  042440  051122
2212  005126  051117     000
2213  005131     112  046525  020120  EM3:    .ASCIZ  /JUMP ERROR/
2214  005136  051105  047522  000122
2215  005144  051103  046501  045040  EM4:    .ASCIZ  /CRAM JUMP TEST FAULT/
2216  005152  046525  020120  042524
2217  005160  052123  043040  052501
2218  005166  052114     000
2219  005171     111  050117  046440  EM5:    .ASCIZ  /IOP MAIN MEMORY TEST/
2220  005176  044501  020116  042515
2221  005204  047515  054522  052040
2222  005212  051505  000124
2223  005216  047511  020120  040515  EM6:    .ASCIZ  /IOP MAR TEST/
2224  005224  020122  042524  052123
2225  005232     000
2226  005233     102  020122  044522  EM7:    .ASCIZ  /BR RIGHT SHIFT ERROR/
2227  005240  044107  020124  044123
2228  005246  043111  020124  051105
2229  005254  047522  000122
2230  005260  040515  020122  052504  EM10:   .ASCIZ  /MAR DUAL ADDRESSING ERROR/
2231  005266  046101  040440  042104
2232  005274  042522  051523  047111
2233  005302  020107  051105  047522
2234  005310  000122
2235  005312  052512  050115  043040  EM11:   .ASCIZ  /JUMP FIELD ERROR/
2236  005320  042511  042114  042440
2237  005326  051122  051117     000
2238  005333     112  046525  020120  EM12:   .ASCIZ  /JUMP TEST ERROR/
2239  005340  042524  052123  042440
2240  005346  051122  051117     000
2241  005353     103  047117  044504  EM16:   .ASCIZ  /CONDITION CODE TESTING,Z & C/
```

```
2242  005360  044524  047117  041440
2243  005366  042117  020105  042524
2244  005374  052123  047111  026107
2245  005402  020132  020046  000103
2246  005410  046103  041517  020113  EMB1:   .ASCIZ  /CLOCK TIME TOO FAST/
2247  005416  044524  042515  052040
2248  005424  047517  043040  051501
2249  005432  000124
2250  005434                          EM35:
2251  005434  047506  041522  020105  EM17:   .ASCIZ  /FORCE POWER FAIL ERROR/
2252  005442  047520  042527  020122
2253  005450  040506  046111  042440
2254  005456  051122  051117     000
2255  005463     111  052502  025123  EM27:   .ASCIZ  "IBUS* WRITE/READ ERROR"
2256  005470  053440  044522  042524
2257  005476  051057  040505  020104
2258  005504  051105  047522  000122
2259
2260  005512  041111  051525  047457  EM29:   .ASCIZ  'IBUS/OBUS WRITE/READ ERROR'
2261  005520  052502  020123  051127
2262  005526  052111  027505  042522
2263  005534  042101  042440  051122
2264  005542  051117     000
2265
2266  005545     120  046507  041440  EMB50:  .ASCIZ  'PGM CLOCK WOULD NOT CLEAR'
2267  005552  047514  045503  053440
2268  005560  052517  042114  047040
2269  005566  052117  041440  042514
2270  005574  051101     000
2271  005577     120  046507  041440  EMB51:  .ASCIZ  'PGM CLOCK WOULD NOT SET'
2272  005604  047514  045503  053440
2273  005612  052517  042114  047040
2274  005620  052117  051440  052105
2275  005626     000
2276  005627     045  022516  025101  STM:    .ASCIZ  '%N%A**************************************************'
2277  005634  025052  025052  025052
2278  005642  025052  025052  025052
2279  005650  025052  025052  025052
2280  005656  025052  025052  025052
2281  005664  025052  025052  025052
2282  005672  025052  025052  025052
2283  005700  025052  025052  025052
2284  005706  025052  025052  025052
2285  005714  025052  025052  025052
2286  005722     000
2287  005723     000                  DH0:    .ASCIZ  //
2288
2289
2290  005724  054105  042520  052103  DH1:    .ASCIZ  /EXPECTED  FOUND  ADDRESS/
2291  005732  042105  020040  047506
2292  005740  047125  020104  040440
2293  005746  042104  042522  051523
2294  005754     000
2295  005755     105  050130  041505  DH2:    .ASCIZ  /EXPECTED  FOUND/
2296  005762  042524  020104  043040
2297  005770  052517  042116     000
```

```
2298  005775     106  047522  020115  DH3:    .ASCIZ  /FROM ADDR  TO ADDR  BAD ADDR/
2299  006002  042101  051104  020040
2300  006010  047524  040440  042104
2301  006016  020122  041040  042101
2302  006024  040440  042104  000122
2303
2304
2305                                  .EVEN
2306
2307
2308
2309                                  ;-------------------------------------------------------------------
2310                                  ; MACRO'S NEEDED TO REPORT ERRORS
2311                                  ;-------------------------------------------------------------------
2312
2313                                  .MACRO  MDT1
2314                                          PRINTB  #TFM1,$REG2,$REG4,$REG0
2315                                  .ENDM
2316
2317                                  .MACRO  MDT2
2318                                          PRINTB  #TFM1,$REG5,$REG4,$REG2
2319                                  .ENDM
2320
2321                                  .MACRO  MDT3
2322                                          PRINTB  #TFM2,$REG5,$REG4
2323                                  .ENDM
2324
2325                                  .MACRO  MDT4
2326                                          PRINTB  #TFM3,$REG5,$REG4,FLAG
2327                                  .ENDM
2328
2329                                  .MACRO  MDT5
2330                                          PRINTB  #TFM3,$REG5,$REG4,$REG2
2331                                  .ENDM
2332
2333                                  .MACRO  MDT0
2334                                  .ENDM
2335                                  .MACRO  MDT6
2336                                          PRINTB  #TFM4,$REG2,$REG4
2337                                  .ENDM
2338
2339                                  .MACRO  MDT7
2340                                          PRINTB  #TFM4,$REG5,$REG4
```

```
2341                                    ENDM
2342                                    .MACRO  MDT8
2343                                            PRINTB #TFM5,FADR,$REG5,$REG4
2344                                    .ENDM
```

```
2345
2346                            .MACRO  $MD     ERRNN   ERNB    ERHM    ERFM
2347                                    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                                  $MD     1,1,1,1
006032                          ERR1::
006032  012746  005C56                  MOV     #EM1,-(SP)
006036  012746  005046                  MOV     #FM1,-(SP)
006042  012746  000002                  MOV     #2,-(SP)
006046  010600                          MOV     SP,R0
006050  104414                          TRAP    C$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
```

```
2404  006224  062706  000006                  ADD     #6,SP
2405  006230  013746  002430                  MOV     $REG0,-(SP)
2406  006234  013746  002420                  MOV     $REG4,-(SP)
2407  006240  013746  002424                  MOV     $REG2,-(SP)
2408  006244  012746  003666                  MOV     #TFM1,-(SP)
2409  006250  012746  000004                  MOV     #4,-(SP)
2410  006254  010600                          MOV     SP,R0
2411  006256  104414                          TRAP    C$PNTB
2412  006260  062706  000012                  ADD     #12,SP
2413  006264  012746  005627                  MOV     #STM,-(SP)
2414  006270  012746  000001                  MOV     #1,-(SP)
2415  006274  010600                          MOV     SP,R0
2416  006276  104414                          TRAP    C$PNTB
2417  006300  062706  000004                  ADD     #4,SP
2418  006304                          L10004:
2419  006304  104423                          TRAP    C$MSG
2420  006306                                  $MD     3,1,1,2
2421  006306                          ERR3::
2422  006306  012746  005056                  MOV     #EM1,-(SP)
2423  006312  012746  005046                  MOV     #FM1,-(SP)
2424  006316  012746  000002                  MOV     #2,-(SP)
2425  006322  010600                          MOV     SP,R0
2426  006324  104414                          TRAP    C$PNTB
2427  006326  062706  000006                  ADD     #6,SP
2428  006332  012746  005724                  MOV     #DH1,-(SP)
2429  006336  012746  005046                  MOV     #FM1,-(SP)
2430  006342  012746  000002                  MOV     #2,-(SP)
2431  006346  010600                          MOV     SP,R0
2432  006350  104414                          TRAP    C$PNTB
2433  006352  062706  000006                  ADD     #6,SP
2434  006356  013746  002424                  MOV     $REG2,-(SP)
2435  006362  013746  002420                  MOV     $REG4,-(SP)
2436  006366  013746  002416                  MOV     $REG5,-(SP)
2437  006372  012746  003666                  MOV     #TFM1,-(SP)
2438  006376  012746  000004                  MOV     #4,-(SP)
2439  006402  010600                          MOV     SP,R0
2440  006404  104414                          TRAP    C$PNTB
2441  006406  062706  000012                  ADD     #12,SP
2442  006412  012746  005627                  MOV     #STM,-(SP)
2443  006416  012746  000001                  MOV     #1,-(SP)
2444  006422  010600                          MOV     SP,R0
2445  006424  104414                          TRAP    C$PNTB
2446  006426  062706  000004                  ADD     #4,SP
2447  006432                          L10005:
2448  006432  104423                          TRAP    C$MSG
2449  006434                                  $MD     4,3,2,3
2450  006434                          ERR4::
2451  006434  012746  005131                  MOV     #EM3,-(SP)
2452  006440  012746  005046                  MOV     #FM1,-(SP)
2453  006444  012746  000002                  MOV     #2,-(SP)
2454  006450  010600                          MOV     SP,R0
2455  006452  104414                          TRAP    C$PNTB
2456  006454  062706  000006                  ADD     #6,SP
2457  006460  012746  005755                  MOV     #DH2,-(SP)
2458  006464  012746  005046                  MOV     #FM1,-(SP)
2459  006470  012746  000002                  MOV     #2,-(SP)
```

```
 2460  006474  010600                        MOV     SP,R0
 2461  006476  104414                        TRAP    C$PNTB
 2462  006500  062706  000006                ADD     #6,SP
 2463  006504  013746  002420                MOV     $REG4,-(SP)
 2464  006510  013746  002416                MOV     $REG5,-(SP)
 2465  006514  012746  003712                MOV     #TFM2,-(SP)
 2466  006520  012746  000003                MOV     #3,-(SP)
 2467  006524  010600                        MOV     SP,R0
 2468  006526  104414                        TRAP    C$PNTB
 2469  006530  062706  000010                ADD     #10,SP
 2470  006534  012746  005627                MOV     #STM,-(SP)
 2471  006540  012746  000001                MOV     #1,-(SP)
 2472  006544  010600                        MOV     SP,R0
 2473  006546  104414                        TRAP    C$PNTB
 2474  006550  062706  000004                ADD     #4,SP
 2475  006554                        L10006:
 2476  006554  104423                        TRAP    C$MSG
```

```
2477  006556                                        $MD     5,4,2,3
2478  006556                                ERR5::
2479  006556  012746  005144                        MOV     #EM4,-(SP)
2480  006562  012746  005046                        MOV     #FM1,-(SP)
2481  006566  012746  000002                        MOV     #2,-(SP)
2482  006572  010600                                MOV     SP,R0
2483  006574  104414                                TRAP    C$PNTB
2484  006576  062706  000006                        ADD     #6,SP
2485  006602  012746  005755                        MOV     #DH2,-(SP)
2486  006606  012746  005046                        MOV     #FM1,-(SP)
2487  006612  012746  000002                        MOV     #2,-(SP)
2488  006616  010600                                MOV     SP,R0
2489  006620  104414                                TRAP    C$PNTB
2490  006622  062706  000006                        ADD     #6,SP
2491  006626  013746  002420                        MOV     $REG4,-(SP)
2492  006632  013746  002416                        MOV     $REG5,-(SP)
2493  006636  012746  003712                        MOV     #TFM2,-(SP)
2494  006642  012746  000003                        MOV     #3,-(SP)
2495  006646  010600                                MOV     SP,R0
2496  006650  104414                                TRAP    C$PNTB
2497  006652  062706  000010                        ADD     #10,SP
2498  006656  012746  005627                        MOV     #STM,-(SP)
2499  006662  012746  000001                        MOV     #1,-(SP)
2500  006666  010600                                MOV     SP,R0
2501  006670  104414                                TRAP    C$PNTB
2502  006672  062706  000004                        ADD     #4,SP
2503  006676                                L10007:
2504  006676  104423                                TRAP    C$MSG
```

```
2505  006700                                          $MD     6,5,1,4
2506  006700                                  ERR6::
2507  006700  012746  005171                          MOV     #EM5,-(SP)
2508  006704  012746  005046                          MOV     #FM1,-(SP)
2509  006710  012746  000002                          MOV     #2,-(SP)
2510  006714  010600                                  MOV     SP,R0
2511  006716  104414                                  TRAP    C$PNTB
2512  006720  062706  000006                          ADD     #6,SP
2513  006724  012746  005724                          MOV     #DH1,-(SP)
2514  006730  012746  005046                          MOV     #FM1,-(SP)
2515  006734  012746  000002                          MOV     #2,-(SP)
2516  006740  010600                                  MOV     SP,R0
2517  006742  104414                                  TRAP    C$PNTB
2518  006744  062706  000006                          ADD     #6,SP
2519  006750  013746  002406                          MOV     FLAG,-(SP)
2520  006754  013746  002420                          MOV     $REG4,-(SP)
2521  006760  013746  002416                          MOV     $REG5,-(SP)
2522  006764  012746  003730                          MOV     #TFM3,-(SP)
2523  006770  012746  000004                          MOV     #4,-(SP)
2524  006774  010600                                  MOV     SP,R0
2525  006776  104414                                  TRAP    C$PNTB
2526  007000  062706  000012                          ADD     #12,SP
2527  007004  012746  005627                          MOV     #STM,-(SP)
2528  007010  012746  000001                          MOV     #1,-(SP)
2529  007014  010600                                  MOV     SP,R0
2530  007016  104414                                  TRAP    C$PNTB
2531  007020  062706  000004                          ADD     #4,SP
2532  007024                                  L10010:
2533  007024  104423                                  TRAP    C$MSG
2534  007026                                          $MD     7,6,1,5
2535  007026                                  ERR7::
2536  007026  012746  005216                          MOV     #EM6,-(SP)
2537  007032  012746  005046                          MOV     #FM1,-(SP)
2538  007036  012746  000002                          MOV     #2,-(SP)
2539  007042  010600                                  MOV     SP,R0
2540  007044  104414                                  TRAP    C$PNTB
2541  007046  062706  000006                          ADD     #6,SP
2542  007052  012746  005724                          MOV     #DH1,-(SP)
2543  007056  012746  005046                          MOV     #FM1,-(SP)
2544  007062  012746  000002                          MOV     #2,-(SP)
2545  007066  010600                                  MOV     SP,R0
2546  007070  104414                                  TRAP    C$PNTB
2547  007072  062706  000006                          ADD     #6,SP
2548  007076  013746  002424                          MOV     $REG2,-(SP)
2549  007102  013746  002420                          MOV     $REG4,-(SP)
2550  007106  013746  002416                          MOV     $REG5,-(SP)
2551  007112  012746  003730                          MOV     #TFM3,-(SP)
2552  007116  012746  000004                          MOV     #4,-(SP)
2553  007122  010600                                  MOV     SP,R0
2554  007124  104414                                  TRAP    C$PNTB
2555  007126  062706  000012                          ADD     #12,SP
2556  007132  012746  005627                          MOV     #STM,-(SP)
2557  007136  012746  000001                          MOV     #1,-(SP)
2558  007142  010600                                  MOV     SP,R0
2559  007144  104414                                  TRAP    C$PNTB
2560  007146  062706  000004                          ADD     #4,SP
```

```
2561  007152                          L10011:
2562  007152  104423                          TRAP    C$MSG
2563  007154                          $MD     10,7,2,3
2564  007154                          ERR10::
2565  007154  012746  005233                  MOV     #EM7,-(SP)
2566  007160  012746  005046                  MOV     #FM1,-(SP)
2567  007164  012746  000002                  MOV     #2,-(SP)
2568  007170  010600                          MOV     SP,R0
2569  007172  104414                          TRAP    C$PNTB
2570  007174  062706  000006                  ADD     #6,SP
2571  007200  012746  005755                  MOV     #DH2,-(SP)
2572  007204  012746  005046                  MOV     #FM1,-(SP)
2573  007210  012746  000002                  MOV     #2,-(SP)
2574  007214  010600                          MOV     SP,R0
2575  007216  104414                          TRAP    C$PNTB
2576  007220  062706  000006                  ADD     #6,SP
2577  007224  013746  002420                  MOV     $REG4,-(SP)
2578  007230  013746  002416                  MOV     $REG5,-(SP)
2579  007234  012746  003712                  MOV     #TFM2,-(SP)
2580  007240  012746  000003                  MOV     #3,-(SP)
2581  007244  010600                          MOV     SP,R0
2582  007246  104414                          TRAP    C$PNTB
2583  007250  062706  000010                  ADD     #10,SP
2584  007254  012746  005627                  MOV     #STM,-(SP)
2585  007260  012746  000001                  MOV     #1,-(SP)
2586  007264  010600                          MOV     SP,R0
2587  007266  104414                          TRAP    C$PNTB
2588  007270  062706  000004                  ADD     #4,SP
2589  007274                          L10012:
2590  007274  104423                          TRAP    C$MSG
2591  007276                          $MD     11,10,2,6
2592  007276                          ERR11::
2593  007276  012746  005260                  MOV     #EM10,-(SP)
2594  007302  012746  005046                  MOV     #FM1,-(SP)
2595  007306  012746  000002                  MOV     #2,-(SP)
2596  007312  010600                          MOV     SP,R0
2597  007314  104414                          TRAP    C$PNTB
2598  007316  062706  000006                  ADD     #6,SP
2599  007322  012746  005755                  MOV     #DH2,-(SP)
2600  007326  012746  005046                  MOV     #FM1,-(SP)
2601  007332  012746  000002                  MOV     #2,-(SP)
2602  007336  010600                          MOV     SP,R0
2603  007340  104414                          TRAP    C$PNTB
2604  007342  062706  000006                  ADD     #6,SP
2605  007346  013746  002420                  MOV     $REG4,-(SP)
2606  007352  013746  002424                  MOV     $REG2,-(SP)
2607  007356  012746  003755                  MOV     #TFM4,-(SP)
2608  007362  012746  000003                  MOV     #3,-(SP)
2609  007366  010600                          MOV     SP,R0
2610  007370  104414                          TRAP    C$PNTB
2611  007372  062706  000010                  ADD     #10,SP
2612  007376  012746  005627                  MOV     #STM,-(SP)
2613  007402  012746  000001                  MOV     #1,-(SP)
2614  007406  010600                          MOV     SP,R0
2615  007410  104414                          TRAP    C$PNTB
2616  007412  062706  000004                  ADD     #4,SP
```

```
2617  007416                          L10013:
2618  007416  104423                          TRAP    C$MSG
2619  007420                                  $MD     12,7,2,7
2620  007420                          ERR12::
2621  007420  012746  005233                  MOV     #EM7,-(SP)
2622  007424  012746  005046                  MOV     #FM1,-(SP)
2623  007430  012746  000002                  MOV     #2,-(SP)
2624  007434  010600                          MOV     SP,R0
2625  007436  104414                          TRAP    C$PNTB
2626  007440  062706  000006                  ADD     #6,SP
2627  007444  012746  005755                  MOV     #DH2,-(SP)
2628  007450  012746  005046                  MOV     #FM1,-(SP)
2629  007454  012746  000002                  MOV     #2,-(SP)
2630  007460  010600                          MOV     SP,R0
2631  007462  104414                          TRAP    C$PNTB
2632  007464  062706  000006                  ADD     #6,SP
2633  007470  013746  002420                  MOV     $REG4,-(SP)
2634  007474  013746  002416                  MOV     $REG5,-(SP)
2635  007500  012746  003755                  MOV     #TFM4,-(SP)
2636  007504  012746  000003                  MOV     #3,-(SP)
2637  007510  010600                          MOV     SP,R0
2638  007512  104414                          TRAP    C$PNTB
2639  007514  062706  000010                  ADD     #10,SP
2640  007520  012746  005627                  MOV     #STM,-(SP)
2641  007524  012746  000001                  MOV     #1,-(SP)
2642  007530  010600                          MOV     SP,R0
2643  007532  104414                          TRAP    C$PNTB
2644  007534  062706  000004                  ADD     #4,SP
2645  007540                          L10014:
2646  007540  104423                          TRAP    C$MSG
2647  007542                                  $MD     13,10,2,3
2648  007542                          ERR13::
2649  007542  012746  005260                  MOV     #EM10,-(SP)
2650  007546  012746  005046                  MOV     #FM1,-(SP)
2651  007552  012746  000002                  MOV     #2,-(SP)
2652  007556  010600                          MOV     SP,R0
2653  007560  104414                          TRAP    C$PNTB
2654  007562  062706  000006                  ADD     #6,SP
2655  007566  012746  005755                  MOV     #DH2,-(SP)
2656  007572  012746  005046                  MOV     #FM1,-(SP)
2657  007576  012746  000002                  MOV     #2,-(SP)
2658  007602  010600                          MOV     SP,R0
2659  007604  104414                          TRAP    C$PNTB
2660  007606  062706  000006                  ADD     #6,SP
2661  007612  013746  002420                  MOV     $REG4,-(SP)
2662  007616  013746  002416                  MOV     $REG5,-(SP)
2663  007622  012746  003712                  MOV     #TFM2,-(SP)
2664  007626  012746  000003                  MOV     #3,-(SP)
2665  007632  010600                          MOV     SP,R0
2666  007634  104414                          TRAP    C$PNTB
2667  007636  062706  000010                  ADD     #10,SP
2668  007642  012746  005627                  MOV     #STM,-(SP)
2669  007646  012746  000001                  MOV     #1,-(SP)
2670  007652  010600                          MOV     SP,R0
2671  007654  104414                          TRAP    C$PNTB
2672  007656  062706  000004                  ADD     #4,SP
```

```
2673  007662                          L10015:
2674  007662  104423                          TRAP    C$MSG
2675  007664                          $MD     14,11,2,6
2676  007664                          ERR14::
2677  007664  012746  005312                  MOV     #EM11,-(SP)
2678  007670  012746  005046                  MOV     #FM1,-(SP)
2679  007674  012746  000002                  MOV     #2,-(SP)
2680  007700  010600                          MOV     SP,R0
2681  007702  104414                          TRAP    C$PNTB
2682  007704  062706  000006                  ADD     #6,SP
2683  007710  012746  005755                  MOV     #DM2,-(SP)
2684  007714  012746  005046                  MOV     #FM1,-(SP)
2685  007720  012746  000002                  MOV     #2,-(SP)
2686  007724  010600                          MOV     SP,R0
2687  007726  104414                          TRAP    C$PNTB
2688  007730  062706  000006                  ADD     #6,SP
2689  007734  013746  002420                  MOV     $REG4,-(SP)
2690  007740  013746  002424                  MOV     $REG2,-(SP)
2691  007744  012746  003755                  MOV     #TFM4,-(SP)
2692  007750  012746  000003                  MOV     #3,-(SP)
2693  007754  010600                          MOV     SP,R0
2694  007756  104414                          TRAP    C$PNTB
2695  007760  062706  000010                  ADD     #10,SP
2696  007764  012746  005627                  MOV     #STM,-(SP)
2697  007770  012746  000001                  MOV     #1,-(SP)
2698  007774  010600                          MOV     SP,R0
2699  007776  104414                          TRAP    C$PNTB
2700  010000  062706  000004                  ADD     #4,SP
2701  010004                          L10016:
2702  010004  104423                          TRAP    C$MSG
```

```
2703  010006                                    $MD     15,12,3,8
2704  010006                              ERR15::
2705  010006  012746  005333                    MOV     #EM12,-(SP)
2706  010012  012746  005046                    MOV     #FM1,-(SP)
2707  010016  012746  000002                    MOV     #2,-(SP)
2708  010022  010600                            MOV     SP,R0
2709  010024  104414                            TRAP    C$PNTB
2710  010026  062706  000006                    ADD     #6,SP
2711  010032  012746  005775                    MOV     #DH3,-(SP)
2712  010036  012746  005046                    MOV     #FM1,-(SP)
2713  010042  012746  000002                    MOV     #2,-(SP)
2714  010046  010600                            MOV     SP,R0
2715  010050  104414                            TRAP    C$PNTB
2716  010052  062706  000006                    ADD     #6,SP
2717  010056  013746  002420                    MOV     $REG4,-(SP)
2718  010062  013746  002416                    MOV     $REG5,-(SP)
2719  010066  013746  002412                    MOV     FADR,-(SP)
2720  010072  012746  003773                    MOV     #TFM5,-(SP)
2721  010076  012746  000004                    MOV     #4,-(SP)
2722  010102  010600                            MOV     SP,R0
2723  010104  104414                            TRAP    C$PNTB
2724  010106  062706  000012                    ADD     #12,SP
2725  010112  012746  005627                    MOV     #STM,-(SP)
2726  010116  012746  000001                    MOV     #1,-(SP)
2727  010122  010600                            MOV     SP,R0
2728  010124  104414                            TRAP    C$PNTB
2729  010126  062706  000004                    ADD     #4,SP
2730  010132                              L10017:
2731  010132  104423                            TRAP    C$MSG
2732  010134                                    $MD     16,16,2,7
2733  010134                              ERR16::
2734  010134  012746  005353                    MOV     #EM16,-(SP)
2735  010140  012746  005046                    MOV     #FM1,-(SP)
2736  010144  012746  000002                    MOV     #2,-(SP)
2737  010150  010600                            MOV     SP,R0
2738  010152  104414                            TRAP    C$PNTB
2739  010154  062706  000006                    ADD     #6,SP
2740  010160  012746  005755                    MOV     #DH2,-(SP)
2741  010164  012746  005046                    MOV     #FM1,-(SP)
2742  010170  012746  000002                    MOV     #2,-(SP)
2743  010174  010600                            MOV     SP,R0
2744  010176  104414                            TRAP    C$PNTB
2745  010200  062706  000006                    ADD     #6,SP
2746  010204  013746  002420                    MOV     $REG4,-(SP)
2747  010210  013746  002416                    MOV     $REG5,-(SP)
2748  010214  012746  003755                    MOV     #TFM4,-(SP)
2749  010220  012746  000003                    MOV     #3,-(SP)
2750  010224  010600                            MOV     SP,R0
2751  010226  104414                            TRAP    C$PNTB
2752  010230  062706  000010                    ADD     #10,SP
2753  010234  012746  005627                    MOV     #STM,-(SP)
2754  010240  012746  000001                    MOV     #1,-(SP)
2755  010244  010600                            MOV     S.,R0
2756  010246  104414                            TRAP    C$PNTB
2757  010250  062706  000004                    ADD     #4,SP
2758  010254                              L10020:
```

2759 010254 104423 TRAP C$MSG
2760

```
2761
2762  010256                                        $MD     17,17,0,0
2763  010256                                ERR17::
2764  010256  012746  005434                        MOV     #EM17,-(SP)
2765  010262  012746  005046                        MOV     #FM1,-(SP)
2766  010266  012746  000002                        MOV     #2,-(SP)
2767  010272  010600                                MOV     SP,R0
2768  010274  104414                                TRAP    C$PNTB
2769  010276  062706  000006                        ADD     #6,SP
2770  010302  012746  005723                        MOV     #DH0,-(SP)
2771  010306  012746  005046                        MOV     #FM1,-(SP)
2772  010312  012746  000002                        MOV     #2,-(SP)
2773  010316  010600                                MOV     SP,R0
2774  010320  104414                                TRAP    C$PNTB
2775  010322  062706  000006                        ADD     #6,SP
2776  010326  012746  005627                        MOV     #STM,-(SP)
2777  010332  012746  000001                        MOV     #1,-(SP)
2778  010336  010600                                MOV     SP,R0
2779  010340  104414                                TRAP    C$PNTB
2780  010342  062706  000004                        ADD     #4,SP
2781  010346                                L10021:
2782  010346  104423                                TRAP    C$MSG
2783  010350                                        $MD     29,29,2,3
2784  010350                                ERR29::
2785  010350  012746  005512                        MOV     #EM29,-(SP)
2786  010354  012746  005046                        MOV     #FM1,-(SP)
2787  010360  012746  000002                        MOV     #2,-(SP)
2788  010364  010600                                MOV     SP,R0
2789  010366  104414                                TRAP    C$PNTB
2790  010370  062706  000006                        ADD     #6,SP
2791  010374  012746  005755                        MOV     #DH2,-(SP)
2792  010400  012746  005046                        MOV     #FM1,-(SP)
2793  010404  012746  000002                        MOV     #2,-(SP)
2794  010410  010600                                MOV     SP,R0
2795  010412  104414                                TRAP    C$PNTB
2796  010414  062706  000006                        ADD     #6,SP
2797  010420  013746  002420                        MOV     $REG4,-(SP)
2798  010424  013746  002416                        MOV     $REG5,-(SP)
2799  010430  012746  003712                        MOV     #TFM2,-(SP)
2800  010434  012746  000003                        MOV     #3,-(SP)
2801  010440  010600                                MOV     SP,R0
2802  010442  104414                                TRAP    C$PNTB
2803  010444  062706  000010                        ADD     #10,SP
2804  010450  012746  005627                        MOV     #STM,-(SP)
2805  010454  012746  000001                        MOV     #1,-(SP)
2806  010460  010600                                MOV     SP,R0
2807  010462  104414                                TRAP    C$PNTB
2808  010464  062706  000004                        ADD     #4,SP
2809  010470                                L10022:
2810  010470  104423                                TRAP    C$MSG
2811  010472                                        $MD     35,35,2,3
2812  010472                                ERR35::
2813  010472  012746  005434                        MOV     #EM35,-(SP)
2814  010476  012746  005046                        MOV     #FM1,-(SP)
2815  010502  012746  000002                        MOV     #2,-(SP)
2816  010506  010600                                MOV     SP,R0
```

```
 2817  010510  104414                          TRAP    C$PNTB
 2818  010512  062706  000006                  ADD     #6,SP
 2819  010516  012746  005755                  MOV     #DH2,-(SP)
 2820  010522  012746  005046                  MOV     #FM1,-(SP)
 2821  010526  012746  000002                  MOV     #2,-(SP)
 2822  010532  010600                          MOV     SP,R0
 2823  010534  104414                          TRAP    C$PNTB
 2824  010536  062706  000006                  ADD     #6,SP
 2825  010542  013746  002420                  MOV     $REG4,-(SP)
 2826  010546  013746  002416                  MOV     $REG5,-(SP)
 2827  010552  012746  003712                  MOV     #TFM2,-(SP)
 2828  010556  012746  000003                  MOV     #3,-(SP)
 2829  010562  010600                          MOV     SP,R0
 2830  010564  104414                          TRAP    C$PNTB
 2831  010566  062706  000010                  ADD     #10,SP
 2832  010572  012746  005627                  MOV     #STM,-(SP)
 2833  010576  012746  000001                  MOV     #1,-(SP)
 2834  010602  010600                          MOV     SP,R0
 2835  010604  104414                          TRAP    C$PNTB
 2836  010606  062706  000004                  ADD     #4,SP
 2837  010612                          L10023:
 2838  010612  104423                          TRAP    C$MSG
 2839
 2840  010614                          BGNMSG  ERR36
 2841  010614                          ERR36::
 2842  010614                          PRINTB  #STM
 2843  010614  012746  005627                  MOV     #STM,-(SP)
 2844  010620  012746  000001                  MOV     #1,-(SP)
 2845  010624  010600                          MOV     SP,R0
 2846  010626  104414                          TRAP    C$PNTB
 2847  010630  062706  000004                  ADD     #4,SP
 2848  010634                          ENDMSG
 2849  010634                          L10024:
 2850  010634  104423                          TRAP    C$MSG
 2851
 2852  010636                          BGNMSG  ERR40
 2853  010636                          ERR40::
 2854  010636                          PRINTF  #TFM40,R2
 2855  010636  010246                          MOV     R2,-(SP)
 2856  010640  012746  004263                  MOV     #TFM40,-(SP)
 2857  010644  012746  000002                  MOV     #2,-(SP)
 2858  010650  010600                          MOV     SP,R0
 2859  010652  104417                          TRAP    C$PNTF
 2860  010654  062706  000006                  ADD     #6,SP
 2861  010660                          PRINTB  #STM
 2862  010660  012746  005627                  MOV     #STM,-(SP)
 2863  010664  012746  000001                  MOV     #1,-(SP)
 2864  010670  010600                          MOV     SP,R0
 2865  010672  104414                          TRAP    C$PNTB
 2866  010674  062706  000004                  ADD     #4,SP
 2867  010700                          ENDMSG
 2868  010700                          L10025:
 2869  010700  104423                          TRAP    C$MSG
 2870  010702                          BGNMSG  ERR41
 2871  010702                          ERR41::
 2872  010702                          PRINTF  #TFM41
```

```
2873  010702  012746  004105                  MOV     #TFM41,-(SP)
2874  010706  012746  000001                  MOV     #1,-(SP)
2875  010712  010600                          MOV     SP,R0
2876  010714  104417                          TRAP    C$PNTF
2877  010716  062706  000004                  ADD     #4,SP
2878  010722                          PRINTB  #STM
2879  010722  012746  005627                  MOV     #STM,-(SP)
2880  010726  012746  000001                  MOV     #1,-(SP)
2881  010732  010600                          MOV     SP,R0
2882  010734  104414                          TRAP    C$PNTB
2883  010736  062706  000004                  ADD     #4,SP
2884  010742                          ENDMSG
2885  010742                          L10026:
2886  010742  104423                          TRAP    C$MSG
2887  010744                          BGNMSG  ERR42
2888  010744                          ERR42::
2889  010744                          PRINTF  #TFM42
2890  010744  012746  004174                  MOV     #TFM42,-(SP)
2891  010750  012746  000001                  MOV     #1,-(SP)
2892  010754  010600                          MOV     SP,R0
2893  010756  104417                          TRAP    C$PNTF
2894  010760  062706  000004                  ADD     #4,SP
2895  010764                          PRINTB  #STM
2896  010764  012746  005627                  MOV     #STM,-(SP)
2897  010770  012746  000001                  MOV     #1,-(SP)
2898  010774  010600                          MOV     SP,R0
2899  010776  104414                          TRAP    C$PNTB
2900  011000  062706  000004                  ADD     #4,SP
2901  011004                          ENDMSG
2902  011004                          L10027:
2903  011004  104423                          TRAP    C$MSG
2904
2905  011006                          BGNMSG  ERR43
2906  011006                          ERR43::
2907  011006                          PRINTF  #TFM43,R5,R4
2908  011006  010446                          MOV     R4,-(SP)
2909  011010  010546                          MOV     R5,-(SP)
2910  011012  012746  004344                  MOV     #TFM43,-(SP)
2911  011016  012746  000003                  MOV     #3,-(SP)
2912  011022  010600                          MOV     SP,R0
2913  011024  104417                          TRAP    C$PNTF
2914  011026  062706  000010                  ADD     #10,SP
2915  011032                          PRINTB  #STM
2916  011032  012746  005627                  MOV     #STM,-(SP)
2917  011036  012746  000001                  MOV     #1,-(SP)
2918  011042  010600                          MOV     SP,R0
2919  011044  104414                          TRAP    C$PNTB
2920  011046  062706  000004                  ADD     #4,SP
2921  011052                          ENDMSG
2922  011052                          L10030:
2923  011052  104423                          TRAP    C$MSG
2924  011054                          BGNMSG  ERR44
2925  011054                          ERR44::
2926  011054                          PRINTF  #TFM44,#TMMC,R4
2927  011054  010446                          MOV     R4,-(SP)
2928  011056  012746  005010                  MOV     #TMMC,-(SP)
```

```
2929  011062  012746  004427                MOV     #TFM44,-(SP)
2930  011066  012746  000003                MOV     #3,-(SP)
2931  011072  010600                         MOV     SP,R0
2932  011074  104417                         TRAP    C$PNTF
2933  011076  062706  000010                ADD     #10,SP
2934  011102                          PRINTB  #STM
2935  011102  012746  005627                MOV     #STM,-(SP)
2936  011106  012746  000001                MOV     #1,-(SP)
2937  011112  010600                         MOV     SP,R0
2938  011114  104414                         TRAP    C$PNTB
2939  011116  062706  000004                ADD     #4,SP
2940  011122                          ENDMSG
2941  011122                          L10031:
2942  011122  104423                         TRAP    C$MSG
2943  011124                          BGNMSG  ERR45
2944  011124                          ERR45::
2945  011124                          PRINTF  #TFM45,#TMMC,R4
2946  011124  010446                         MOV     R4,-(SP)
2947  011126  012746  005010                MOV     #TMMC,-(SP)
2948  011132  012746  004466                MOV     #TFM45,-(SP)
2949  011136  012746  000003                MOV     #3,-(SP)
2950  011142  010600                         MOV     SP,R0
2951  011144  104417                         TRAP    C$PNTF
2952  011146  062706  000010                ADD     #10,SP
2953  011152                          PRINTB  #TFM45A
2954  011152  012746  004524                MOV     #TFM45A,-(SP)
2955  011156  012746  000001                MOV     #1,-(SP)
2956  011162  010600                         MOV     SP,R0
2957  011164  104414                         TRAP    C$PNTB
2958  011166  062706  000004                ADD     #4,SP
2959  011172                          PRINTB  #STM
2960  011172  012746  005627                MOV     #STM,-(SP)
2961  011176  012746  000001                MOV     #1,-(SP)
2962  011202  010600                         MOV     SP,R0
2963  011204  104414                         TRAP    C$PNTB
2964  011206  062706  000004                ADD     #4,SP
2965  011212                          ENDMSG
2966  011212                          L10032:
2967  011212  104423                         TRAP    C$MSG
2968  011214                          BGNMSG  ERR46
2969  011214                          ERR46::
2970  011214                          PRINTF  #TFM46,$GDDAT,R4
2971  011214  010446                         MOV     R4,-(SP)
2972  011216  013746  002452                MOV     $GDDAT,-(SP)
2973  011222  012746  004647                MOV     #TFM46,-(SP)
2974  011226  012746  000003                MOV     #3,-(SP)
2975  011232  010600                         MOV     SP,R0
2976  011234  104417                         TRAP    C$PNTF
2977  011236  062706  000010                ADD     #10,SP
2978  011242                          PRINTB  #STM
2979  011242  012746  005627                MOV     #STM,-(SP)
2980  011246  012746  000001                MOV     #1,-(SP)
2981  011252  010600                         MOV     SP,R0
2982  011254  104414                         TRAP    C$PNTB
2983  011256  062706  000004                ADD     #4,SP
2984  011262                          ENDMSG
```

```
 2985  011262                          L10033:
 2986  011262  104423                          TRAP    C$MSG
 2987
 2988  011264                          BGNMSG  ERR47
 2989  011264                          ERR47::
 2990  011264                          PRINTF  #TFM47,R5,R4
 2991  011264  010446                          MOV     R4,-(SP)
 2992  011266  010546                          MOV     R5,-(SP)
 2993  011270  012746  004733                  MOV     #TFM47,-(SP)
 2994  011274  012746  000003                  MOV     #3,-(SP)
 2995  011300  010600                          MOV     SP,R0
 2996  011302  104417                          TRAP    C$PNTF
 2997  011304  062706  000010                  ADD     #10,SP
 2998  011310                          PRINTB  #STM
 2999  011310  012746  005627                  MOV     #STM,-(SP)
 3000  011314  012746  000001                  MOV     #1,-(SP)
 3001  011320  010600                          MOV     SP,R0
 3002  011322  104414                          TRAP    C$PNTB
 3003  011324  062706  000004                  ADD     #4,SP
 3004  011330                          ENDMSG
 3005  011330                          L10034:
 3006  011330  104423                          TRAP    C$MSG
 3007
```

```
                                .SBTTL  REPORT CODING SECTION
```

```
3011                                    ;**
```

```
3012                                    ; THE REPORT CODING SECTION CONTAINS THE
3013                                    ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
3014                                    ;--
3015
3016  011332                                    BGNRPT
3017  011332                            L$RPT::
3018
3019
3020  011332                                    EXIT    RPT
3021  011332  000167                            .WORD   J$JMP
3022  011334  000000                            .WORD   L10035-2-.
3023
```

```
3024
3025  011336                          ENDRPT
3026  011336                  L10035:
3027  011336  104425                  TRAP    C$RPT
3028
```

```
3029
3030
```

```
3031
3032
```

```
 3033                                   .SBTTL  INITIALIZE SECTION
```

```
3034
3035                                    ;////////////////////////////////////////////////////////////////////////
3036                                    ;/ THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
3037                                    ;/ AT THE BEGINNING OF EACH PASS.
3038                                    ;////////////////////////////////////////////////////////////////////////
3039
3040  011340                                    BGNINIT
3041  011340                            L$INIT::
3042
3043                                    ;INITIALIZE SUBROUTINE STACK
3044  011340  012705  002730                    MOV     #SSTACK,R5
3045                                    ;STORE BASE LEVEL PROGRAM STACK POINTER
3046  011344  010637  002344                    MOV     SP,PSTACK
3047  011350  005737  002462                    TST     FTIME
3048  011354  001011                            BNE     1$
3049  011356  013737  000004  002464           MOV     @#4,SAVE4
3050  011364  013737  000006  002466           MOV     @#6,SAVE6
3051  011372  012737  000001  002462           MOV     #1,FTIME
3052  011400  013737  002464  000004  1$:      MOV     SAVE4,@#4
3053  011406  013737  002466  000006           MOV     SAVE6,@#6
3054
3055                                    ;SEE IF PROGRAM JUST STARTED, BR IF YES
3056  011414                                    READEF  #EF.START
3057  011414  012700  000040                    MOV     #EF.START,R0
3058  011420  104447                            TRAP    C$REFG
3059  011422                                    BCOMPLETE       NEWST
3060  011422  103414                            BCS     NEWST
3061                                    ;SEE IF THIS IS A NEW PASS, BR IF YES
3062  011424                                    READEF  #EF.NEW
3063  011424  012700  000035                    MOV     #EF.NEW,R0
3064  011430  104447                            TRAP    C$REFG
3065  011432                                    BCOMPLETE       NEWST
3066  011432  103410                            BCS     NEWST
3067                                    ;SEE IF PROGRAM WAS JUST CONTINUED
3068  011434                                    READEF  #EF.CONTINUE
3069  011434  012700  000036                    MOV     #EF.CONTINUE,R0
3070  011440  104447                            TRAP    C$REFG
3071  011442                                    BCOMPLETE       ENDIT
3072  011442  103576                            BCS     ENDIT
3073                                    ;SEE IF PROGRAM JUST RESTARTED, BR IF NOT
3074  011444                                    READEF  #EF.RESTART
3075  011444  012700  000037                    MOV     #EF.RESTART,R0
3076  011450  104447                            TRAP    C$REFG
3077  011452                                    BNCOMPLETE      GETPRM
3078  011452  103003                            BCC     GETPRM
3079
3080  011454                            NEWST:
3081                                    ;RESET LOGICAL DEVICE TO -1
3082  011454  012737  177777  002342            MOV     #-1,LOGDEV
3083
3084                                    ;GET UNIBUS ADRS, VECTOR, PRIORITY LEVEL, LINE UNIT,SWITCH
3085                                    ;    PACKS, TEST CONNECTOR INFO. FOR THIS M8200,4,7 (CURRENT LOGICAL
3086                                    ;    DEVICE).
3087  011462                            GETPRM:
3088  011462  005237  002342                    INC     LOGDEV
3089  011466  023737  002342  002012           CMP     LOGDEV,L$UNIT
```

```
3090  011474  002367                          BGE     NEWST
3091  011476                                  GPHARD  LOGDEV,R1
3092  011476  013700  002342                  MOV     LOGDEV,R0
3093  011502  104442                          TRAP    C$GPHRD
3094  011504  010001                          MOV     R0,R1
3095  011506                                  BNCOMPLETE      GETPRM
3096  011506  103365                          BCC     GETPRM
3097  011510  012137  002414                  MOV     (R1)+,WTYPE
3098                                  ;GET ADDRESS OF M8200,4,7
3099  011514  011137  002516                  MOV     (R1),KMCSR
3100                                  ;GET POINTER TO M8200,4,7 CSR HI BYTE
3101  011520  011137  002520                  MOV     (R1),KMCSRH
3102  011524  005237  002520                  INC     KMCSRH
3103                                  ;GET POINTER TO M8200,4,7 CTL OUT REG
3104  011530  011137  002522                  MOV     (R1),KMCTL
3105  011534  062737  000002  002522          ADD     #2,KMCTL
3106                                  ;GET POINTER TO M8200,4,7 PORT REG - SEL 4
3107  011542  011137  002524                  MOV     (R1),KMPO4
3108  011546  062737  000004  002524          ADD     #4,KMPO4
3109                                  ;GET POINTER TO M8200,4,7 PORT REG - SEL 6
3110  011554  012137  002526                  MOV     (R1)+,KMPO6
3111  011560  062737  000006  002526          ADD     #6,KMPO6
3112                                  ;GET POINTER TO RCV VECTOR
3113  011566  011137  002506                  MOV     (R1),KMRVEC
3114                                  ;GET POINTER TO RCV PRIORITY LEVEL
3115  011572  011137  002510                  MOV     (R1),KMRLVL
3116  011576  062737  000002  002510          ADD     #2,KMRLVL
3117                                  ;GET POINTER TO TX VECTOR
3118  011604  011137  002512                  MOV     (R1),KMTVEC
3119  011610  062737  000004  002512          ADD     #4,KMTVEC
3120                                  ;GET POINTER TO TX PRIORITY LEVEL
3121  011616  011137  002514                  MOV     (R1),KMTLVL
3122  011622  062737  000006  002514          ADD     #6,KMTLVL
3123                                  ;PUT VECTOR INTO STAT1
3124  011630  016137  000020  002472          MOV     20(R1),RUNINH
3125  011636  012137  002500                  MOV     (R1)+,STAT1
3126                                  ;PUT PRIORITY INTO STAT1
3127  011642  052137  002500                  BIS     (R1)+,STAT1
3128                                  ;SEE IF NO LINE UNIT, SET BIT IF YES
3129  011646  005711                          TST     (R1)
3130  011650  001004                          BNE     50000$
3131  011652  052737  010000  002500          BIS     #BIT12,STAT1
3132  011660  000416                          BR      4$
3133  011662                                  50000$:
3134                                  ;SEE IF M8201 LINE UNIT, SET BIT IF YES
3135  011662  021127  000001                  CMP     (R1),#1
3136  011666  001001                          BNE     50001$
3137  011670  000412                          BR      4$
3138  011672                                  50001$:
3139                                  ;SEE IF M8202 LINE UNIT, SET BIT IF YES
3140  011672  021127  000002                  CMP     (R1),#2
3141  011676  001004                          BNE     50002$
3142  011700  052737  020000  002500          BIS     #BIT13,STAT1
3143  011706  000403                          BR      4$
3144  011710                                  50002$:
3145                                  ;SET BIT FOR M8203 LINE UNIT
```

```
3146  011710  052737  100000  002500          BIS     #BIT15,STAT1
3147  011716                                4$:
3148                                        ;SET BIT IN STAT1 FOR TEST CONNECTOR
3149  011716  056137  000006  002500          BIS     6(R1),STAT1
3150  011724  062701  000002                  ADD     #2,R1
3151                                        ;SET SWITCH PACK #1 IN STAT2 LOW BYTE
3152  011730  012137  002502                  MOV     (R1)+,STAT2
3153                                        ;SET SWITCH PACK #2 IN STAT2 HIGH BYTE
3154  011734  111137  002503                  MOVB    (R1),STAT2+1
3155
3156                                        ;INCREMENT LOGICAL UNIT (DEVICE) NUMBER
3157                                        ;       INC     LOGDEV
3158  011740  000240                          NOP
3159  011742  000240                          NOP
3160
3161  011744  012737  002000  002436          MOV     #2000,MEMSZ
3162  011752  005037  002432                  CLR     TYPE
3163  011756  123727  002414  000000          CMPB    WTYPE,#0
3164  011764  001425                          BEQ     ENDIT
3165  011766  123727  002414  000004          CMPB    WTYPE,#4        ;KMC?
3166  011774  001004                          BNE     5$
3167  011776  012737  000001  002432          MOV     #1,TYPE
3168  012004  000415                          BR      ENDIT
3169  012006  012737  007777  002436  5$:     MOV     #7777,MEMSZ
3170  012014  123727  002414  000006          CMPB    WTYPE,#6
3171  012022  001003                          BNE     7$
3172  012024  012737  000001  002432          MOV     #1,TYPE
3173  012032  013737  002472  002470  7$:     MOV     RUNINH,RUNB
3174  012040                          6$:
3175  012040                          ENDIT:
3176  012040                                  ENDINIT
3177  012040                          L10036:
3178  012040  104411                          TRAP    C$INIT
3179
3180                                  .EVEN
3181  012042                                  BGNAUTO
3182  012042                          L$AUTO::
3183                                          ;DEVICE DOES NOT HAVE A "READY"
3184  012042  013701  002516                  MOV     KMCSR,R1        ;R1 CONTAINS BASE M8200,4,7 ADDRESS
3185  012046  012705  000004                  MOV     #4,R5           ;4 REGISTERS TO BE TESTED
3186  012052  012737  012104  000004          MOV     #2$,4           ;SET OUT TIMEOUT TRAP
3187  012060  012737  000240  000006          MOV     #240,6          ;LEVEL 7
3188  012066  005711                  1$:     TST     (R1)            ;REFERENCE DEVICE REGISTERS
3189  012070  000240                          NOP
3190  012072  062701  000002                  ADD     #2,R1           ;NEXT REGISTER
3191  012076  005305                          DEC     R5              ;DEC REGISTER COUNT
3192  012100  001372                          BNE     1$              ;BR IF NOT LAST REGISTER
3193  012102  000405                          BR      3$
3194
3195  012104  062706  000004          2$:     ADD     #4,SP
3196  012110                                  DODU    LOGDEV
3197  012110  013700  002342                  MOV     LOGDEV,R0
3198  012114  104451                          TRAP    C$DODU
3199
3200  012116  013737  002464  000004  3$:     MOV     SAVE4,4
3201  012124  013737  002466  000006          MOV     SAVE6,6
```

```
3202  012132
3203  012132                          ENDAUTO
3204  012132  104461      .  L10037:  TRAP    C$AUTO
3205
```

```
                                .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
```

```
3226                                    .SBTTL  DROP UNIT SECTION
3227
3228                                    ;////////////////////////////////////////////////////////////////////////////
3229                                    ;/ THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
3230                                    ;/ TO NO LONGER BE TESTED.
3231                                    ;////////////////////////////////////////////////////////////////////////////
3232
3233  012140                                    BGNDU
3234  012140                            L$DU::
3235                                    ;ISSUE UNIBUS RESET TO CLEAN UP
3236  012140                                    BRESET
3237  012140  104433                            TRAP    C$RESET
3238  012142                                    ENDDU
3239  012142                            L10041:
3240  012142  104453                            TRAP    C$DU
3241
3242
3243
3244
3245
```

```
                                .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.AUN!!" 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:
012146                                  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   EM0
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 **************
```

```
3321
3322  012256                                BGNTST
3323  012256                                T2::
3324
3325  012256                                        MSTCLR                  ;R1 CONTAINS BASE M8200,4,7 ADDRESS
3326  012262  013701  002516                        MOV     KMCSR,R1        ;MASTER CLEAR M8200,4,7
3327  012266  005011                                CLR     (R1)            ;R1 = M8200,4,7 BASE ADDRESS
3328  012270  012705  052525                        MOV     #52525,R5       ;CLEAR SELO
3329  012274  010561  000004                        MOV     R5,4(R1)        ;START WITH 125
3330  012300                                        ROMCLK                  ;PORT4 125
3331  012300  004537  003044                        JSR     R5,..ROMCLK     ;NEXT WORD IS INSTRUCTION
3332  012304  120500                                120500                  ;CLOCK INSTRUCTION
3333  012306                                        ROMCLK                  ;PORT4 TO BR-REG
3334  012306  004537  003044                        JSR     R5,..ROMCLK     ;NEXT WORD IS INSTRUCTION
3335  012312  061620                                061620                  ;CLOCK INSTRUCTION
3336  012314                                        ROMCLK                  ;BR RSH BR, SHIFT BR RIGHT
3337  012314  004537  003044                        JSR     R5,..ROMCLK     ;NEXT WORD IS INSTRUCTION
3338  012320  061225                                061225                  ;CLOCK INSTRUCTION
3339  012322  006005                                ROR     R5              ;PORT5 BR
3340  012324  005004                                CLR     R4              ;R5 = "EXPECTED"
3341  012326  116104  000005                        MOVB    5(R1),R4        ;R4 = "FOUND"
3342  012332  120504                                CMPB    R5,R4           ;DID BR SHIFT RIGHT ONCE?
3343  012334  001410                                BEQ     1$              ;BR IF YES
3344  012336                                        ERROR   12              ;BR RIGHT SHIFT ERROR
3345  012342  104455                                TRAP    C$ERDF
3346  012344  000014                                .WORD   12
3347  012346  005055                                .WORD   EMC
3348  012350  007420                                .WORD   ERR12
3349
3350  012352                                        ESCAPE TST              ;SHOULD BE 52
3351  012352  104410                                TRAP    C$ESCAPE
3352  012354  000044                                .WORD   L10044-.
3353  012356                                1$:
3354  012356                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION
3355  012356  004537  003044                        JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3356  012362  061620                                061620                  ;BR RSH BR, SHFT BR RIGHT AGAIN
3357  012364                                        ROMCLK                  ;NE4XT WORD IS INSTRUCTION
3358  012364  004537  003044                        JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3359  012370  061225                                061225                  ;PORT5 BR
3360  012372  006005                                ROR     R5              ;R5 = "EXPECTED"
3361  012374  116104  000005                        MOVB    5(R1),R4        ;R4 = "FOUND"
3362  012400  120504                                CMPB    R5,R4           ;DID BR SHIFT RIGHT?
3363  012402  001406                                BEQ     2$              ;BR IF YES
3364  012404                                        ERROR   12              ;BR RIGHT SHIFT ERROR
3365  012410  104455                                TRAP    C$ERDF
3366  012412  000014                                .WORD   12
3367  012414  005055                                .WORD   EMO
3368  012416  007420                                .WORD   ERR12
3369
3370  012420                                        ;S/B 25
3371  012420                                2$:
3372  012420                                ENDTST
3373  012420  104401                        L10044:
3374                                                TRAP    C$ETST
3375  012422                                        BADHEAD
3376                                                ;************** 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$EX!T
012432  000116                          .WORD   L10045-.
012434                                  MYINT
012434  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
                                                                        ;R1 CONTAINS BASE M3200,4,7 ADDRESS
012440  005037  002434                  CLR     MR0                     ;MR0 = CRAM ADDRESS
012444  012702  000001          ADR4:   MOV     #1,R2                   ;R2 = WRITE DATA
012450                          ADR5:
012450                                  BGNSEG
012450  104404                          TRAP    C$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
```

```
3433  012552                                  T4::
3434  012552                                  MACEX
3435
3436  012560  104432                                  ;DO NOT DO TEST IF M8200
3437  012562  000126                          TRAP    C$EXIT
3438  012564                                  .WORD   L10046-.
3439  012564  013701  002516                  MYINT
3440  012570                                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
3441  012574  005037  002434                  MSTCLR                          ;MASTER CLEAR M8200,4,7
3442  012600  012702  000001                  CLR     MR0                     ;MR0 = CRAM ADDRESS
3443  012604                          ADR1:   MOV     #1,R2                   ;R2 = WRITE DATA
3444  012604                          ADR2:
3445  012604  104404                          BGNSEG
3446  012606  005102                          TRAP    C$BSEG
3447  012610  012711  002000          3$:     COM     R2                      ;MAKE IT A FLOATING ZERO
3448  012614  013761  002434  000004          MOV     #BIT10,(R1)             ;SET ROMO
3449  012622  010261  000006                  MOV     MR0,4(R1)               ;WRITE ADDRESS TO SEL4
3450  012626  052711  020000                  MOV     R2,6(R1)                ;LOAD SEL6 WITH WRITE DATA
3451  012632  016104  000006                  BIS     #BIT13,(R1)             ;WRITE SEL6 INTO CRAM
3452  012636  020204                          MOV     6(R1),R4                ;READ CRAM INTO "FOUND"
3453  012640  001410                          CMP     R2,R4                   ;IS DATA CORRECT?
3454  012642                                  BEQ     4$                      ;BR IF OK
3455  012646  104455                          ERROR   1                       ;ERROR
3456  012650  000001                          TRAP    C$ERDF
3457  012652  005055                          .WORD   1
3458  012654  006032                          .WORD   EM0
3459  012656                                  .WORD   ERR1
3460  012656  104410                          ESCAPE SEG
3461  012660  000002                          TRAP    C$ESCAPE
3462  012662                                  .WORD   10000$-.
3463  012662                          4$:     ENDSEG
3464  012662  104405                  10000$:
3465  012664  005102                          TRAP    C$ESEG
3466  012666  000241                          COM     R2                      ;BACK TO FLOATING ONE
3467  012670  006102                          CLC                             ;CLEAR CARRY
3468  012672  001344                          ROL     R2                      ;SHIFT WRITE DATA
3469  012674  005237  002434                  BNE     ADR2                    ;BR IF NOT DONE THIS ADDRESS
3470  012700  023737  002436  002434          INC     MR0                     ;BUMP TO NEXT CRAM ADDRESS
3471  012706  001334                          CMP     MEMSZ,MR0               ;DONE YET?
3472  012710                                  BNE     ADR1                    ;BR IF NO
3473  012710                          5$:
3474  012710                          ENDTST
3475  012710  104401                  L10046:
3476                                          TRAP    C$ETST
3477  012712
3478                                          BADHEAD
3479                                          ;************** TEST 5 **************
3480                                          ;*IOP CRAM DUAL ADDRESSING TEST
3481                                          ;*WRITE EACH ADDRESS INTO ITSELF, READ EACH
3482  012712                                  ;*ADDRESS TO VERIFY CORRECT ADDRESSING
3483                                          BADHEAD
3484                                          ;************** TEST 5 **************
3485  012712
3486  012712                          BGNTST
3487  012712                          T5::
3488                                  MACEX
                                              ;DO NOT DO TEST IF M8200
```

```
3489  012720  104432                          TRAP    C$EXIT
3490  012722  000230                          .WORD   L10047-.
3491  012724                                  MYINT
3492  012724  013701  002516                  MOV     KMCSR,R1            ;RECORD DEVICE ADDR.
3493                                                                      ;R1 CONTAINS BASE M8200,4,7 ADDRESS
3494  012730                                  MSTCLR                      ;MASTER CLEAR M8200,4,7
3495  012734  005037  002434                  CLR     MR0                 ;MR0 =CRAM ADDRESS
3496  012740                                  BGNSEG
3497  012740  104404                          TRAP    C$BSEG
3498  012742  013702  002434          1$:     MOV     MR0,R2              ;SAVE R2 FOR TYPEOUT
3499  012746  012711  002000                  MOV     #BIT10,(R1)         ;SET ROM0
3500  012752  013761  002434  000004          MOV     MR0,4(R1)           ;WRITE ADDRESS TO SEL4
3501  012760  013761  002434  000006          MOV     MR0,6(R1)           ;LOAD SEL6 WITH WRITE DATA
3502  012766  052711  020000                  BIS     #BIT13,(R1)         ;WRITE CRAM
3503  012772                                  SKIP06  15$                 ;IF M8206,SKIP NEXT INSTR.
3504                                          ;GOTO 15$ IF M8206
3505  013002  005061  000006                  CLR     6(R1)               ;CLEAR SEL 6
3506  013006                          15$:
3507  013006  016104  000006                  MOV     6(R1),R4            ;SHOULD READ BACK OWN ADDRESS
3508  013012  023704  002434                  CMP     MR0,R4              ;IS DATA CORRECT?
3509  013016  001410                          BEQ     2$                  ;BR IF YES
3510  013020                                  ERROR   1                   ;DATA ERROR
3511  013024  104455                          TRAP    C$ERDF
3512  013026  000001                          .WORD   1
3513  013030  005055                          .WORD   EM0
3514  013032  006032                          .WORD   ERR1
3515  013034                                  ESCAPE SEG
3516  013034  104410                          TRAP    C$ESCAPE
3517  013036  000002                          .WORD   10000$-.
3518  013040                          2$:     ENDSEG
3519  013040                          10000$:
3520  013040  104405                          TRAP    C$ESEG
3521  013042                                  BGNSEG
3522  013042  104404                          TRAP    C$BSEG
3523  013044  005237  002434                  INC     MR0                 ;BUMP TO NEXT ADDRESS
3524  013050  023737  002436  002434          CMP     MEMSZ,MR0           ;DONE WRITING YET?
3525  013056  001331                          BNE     1$                  ;BR IF NO
3526  013060  005037  002434                  CLR     MR0                 ;RESTART AT ADDRESS 0
3527  013064  013702  002434          3$:     MOV     MR0,R2              ;SAVE R2 FOR TYPEOUT
3528  013070  012711  002000                  MOV     #BIT10,(R1)         ;SET ROM0
3529  013074  013761  002434  000004          MOV     MR0,4(R1)           ;SEL4 = CRAM ADDRESS
3530  013102  016104  000006                  MOV     6(R1),R4            ;READ CRAM INTO "FOUND"
3531  013106  023704  002434                  CMP     MR0,R4              ;IS DATA CORRECT?
3532  013112  001411                          BEQ     4$                  ;BR IF YES
3533  013114                                  ERROR   2                   ;DUAL ADDRESSING ERROR
3534  013120  104455                          TRAP    C$ERDF
3535  013122  000002                          .WORD   2
3536  013124  005055                          .WORD   EM0
3537  013126  006160                          .WORD   ERR2
3538  013130                                  ESCAPE SEG
3539  013130  104410                          TRAP    C$ESCAPE
3540  013132  000002                          .WORD   10001$-.
3541  013134                                  ENDSEG
3542  013134                          10001$:
3543  013134  104405                          TRAP    C$ESEG
3544  013136                          4$:                                 ;LOOP TO 3$ IF SW09=1
```

```
3545  013136  005237  002434                  INC     MR0             ;BUMP TO NEXT ADDRESS
3546  013142  023737  002436  002434          CMP     MEMSZ,MR0       ;DONE WRITING YET?
3547  013150  001345                          BNE     3$              ;BR IF NO
3548  013152                          5$:
3549  013152                          ENDTST
3550  013152                          L10047:
3551  013152  104401                          TRAP    C$ETST
3552
3553
3554  013154                                  BADHEAD
3555                                          ;************** TEST 6 **************
3556                                          ;*IOP MAIN MEMORY TEST
3557                                          ;*FLOAT A 1 THROUGH ALL MAIN MEMORY LOCATIONS
3558  013154                                  BADHEAD
3559                                          ;************** TEST 6 **************
3560
```

```
3561  013154                                BGNTST
3562  013154                                T6::
3563  013154                                      MYINT
3564  013154  013701  002516                      MOV     KMCSR,R1          ;RECORD DEVICE ADDR.
3565                                                                        ;R1 CONTAINS BASE M8200,4,7 ADDRESS
3566  013160                                      MSTCLR                    ;MASTER CLEAR M8200,4,7
3567  013164  005037  002406                      CLR     FLAG              ;START WITH ADDRESS 0
3568  013170  012737  000001  002434  1$:         MOV     #1,MR0            ;START WITH BIT 0
3569  013176  042737  003777  013232  65$:        BIC     #3777,66$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
```

```
 3570  013204  042737  000037  013240          BIC     #37,68$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
```

```
3571  013212  153737  002406  013252          BISB    FLAG,66$        ;ADD ADDRESS TO INSTRUCTION?
3572  013220  153737  002407  013240          BISB    FLAG+1,68$      ;ADD ADDRESS TO INSTRUCTION
3573  013226                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3574  013226  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3575  013232  010000                  66$:    0*0000
3576  013234                                  ROMCLK
3577  013234  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3578  013240  004000                  68$:    004000                  ;LOAD MAR HI
3579  013242  013761  002434  000004          MOV     MR0,4(R1)       ;WRITE PATTERN IN PORT4
3580  013250                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3581  013250  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3582  013254  122500                          122500                  ;MOVE PORT4 TO MEMORY
3583  013256                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3584  013256  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3585  013262  040620                          040620                  ;MOVE MEMORY TO BR
3586  013264                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3587  013264  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3588  013270  061225                          61225                   ;MOVE BR TO PORT5
3589  013272  013705  002434                  MOV     MR0,R5          ;PUT "EXPECTED" IN R5
3590  013276  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" IN R4
3591  013302  120504                          CMPB    R5,R4           ;DATA CORRECT?
3592  013304  001410                          BEQ     67$             ;BR IF YES
3593  013306                                  ERROR   6               ;DATA ERROR
3594  013312  104455                          TRAP    C$ERDF
3595  013314  000006                          .WORD   6
3596  013316  005055                          .WORD   EM0
3597  013320  006700                          .WORD   ERR6
3598  013322                                  ESCAPE  TST
3599  013322  104410                          TRAP    C$ESCAPE
3600  013324  000030                          .WORD   L10050-.
3601  013326                          67$:                            ;SW09=1?
3602  013326  000241                          CLC                     ;CLEAR CARRY
3603  013330  106137  002434                  ROLB    MR0             ;SHIFT BIT IN MR0
3604  013334  001320                          BNE     65$             ;DONE IF MR0=0
3605  013336                                  BREAK
3606  013336  104422                          TRAP    C$BRK
3607  013340  005237  002406                  INC     FLAG            ;NEXT ADDRESS
3608  013344  023737  002436  002406          CMP     MEMSZ,FLAG      ;LAST ADDRESS?
3609  013352  001306                          BNE     1$              ;BR IF NO
3610  013354                          2$:
3611  013354                          ENDTST
3612  013354                          L10050:
3613  013354  104401                          TRAP    C$ETST
3614
3615  013356                                  BADHEAD
3616                                          ;************** TEST 7 **************
3617                                          ;*IOP MAIN MEMORY TEST
3618                                          ;*FLOAT A 0 THROUGH ALL MAIN MEMORY LOCATIONS
3619  013356                                  BADHEAD
3620                                          ;************** TEST 7 **************
3621
3622  013356                          BGNTST
3623  013356                          T7::
3624  013356                                  MYINT
3625  013356  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
3626                                                                  ;R1 CONTAINS BASE M8200,4,7 ADDRESS
```

```
3627  013362                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
3628  013366  005037  002406                         CLR     FLAG            ;START WITH ADDRESS 0
3629  013372  012737  000001  002434          1$:    MOV     #1,MR0          ;START WITH BIT 0
3630  013400  005137  002434                  64$:    COM     MR0             ;CHANGE TO FLOATING 0
3631  013404  042737  003777  013440          65$:    BIC     #3777,66$       ;CLEAR ADDRESS FIELD OF INSTRUCTION
3632  013412  042737  000037  013446                  BIC     #37,68$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
3633  013420  153737  002406  013440                  BISB    FLAG,66$        ;ADD ADDRESS TO INSTRUCTION
3634  013426  153737  002407  013446                  BISB    FLAG+1,68$      ;ADD ADDRESS TO INSTRUCTION
3635  013434                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3636  013434  004537  003044                         JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3637  013440  010000                          66$:    010000                  ;LOAD MAR LO WITH ADDRESS IN FLAG
3638  013442                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3639  013442  004537  003044                         JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3640  013446  004000                          68$:    004000                  ;LOAD MAR HI
3641  013450  013761  002434  000004                 MOV     MR0,4(R1)       ;WRITE PATTERN IN PORT4
3642  013456                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3643  013456  004537  003044                         JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3644  013462  122500                                  122500                  ;MOVE PORT4 TO MEMORY
3645  013464                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3646  013464  004537  003044                         JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3647  013470  040620                                  040620                  ;MOVE MEMORY TO BR
3648  013472                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3649  013472  004537  003044                         JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3650  013476  061225                                  61225                   ;MOVE BR TO PORT5
3651  013500  013705  002434                         MOV     MR0,R5          ;PUT "EXPECTED" IN R5
3652  013504  116104  000005                         MOVB    5(R1),R4        ;PUT "FOUND" IN R4
3653  013510  120504                                  CMPB    R5,R4           ;DATA CORRECT?
3654  013512  001406                                  BEQ     67$             ;BR IF YES
3655  013514                                          ERROR   6               ;DATA ERROR
3656  013520  104455                                  TRAP    C$ERDF
3657  013522  000006                                  .WORD   6
3658  013524  005055                                  .WORD   EM0
3659  013526  006700                                  .WORD   ERR6
3660  013530                                  67$:    ESCAPE TST
3661  013530  104410                                  TRAP    C$ESCAPE
3662  013532  000034                                  .WORD   L10051-.
3663  013534  005137  002434                         COM     MR0             ;CHANGE TO FLOATING 1
3664  013540  000241                                  CLC                     ;CLEAR CARRY
3665  013542  106137  002434                         ROLB    MR0             ;SHIFT BIT IN MR0
3666  013546  001314                                  BNE     64$             ;DONE IF MR0=0
3667  013550                                          BREAK
3668  013550  104422                                  TRAP    C$BRK
3669  013552  005237  002406                         INC     FLAG            ;NEXT ADDRESS
3670  013556  023737  002436  002406                 CMP     MEMSZ,FLAG      ;LAST ADDRESS?
3671  013564  001302                                  BNE     1$              ;BR IF NO
3672  013566                                  2$:
3673  013566                                  ENDTST
3674  013566                                  L10051:
3675  013566  104401                                  TRAP    C$ETST
3676
3677  013570                                          BADHEAD
3678                                                  ;************** TEST 8 ***************
3679                                                  ;*IOP MAIN MEMORY DUAL ADDRESSING TEST
3680                                                  ;*LOAD EACH MEMORY LOCATION WITH ITS OWN ADDRESS
3681                                                  ;*READ BACK EACH LOCATION TO VERIFY CORRECT ADDRESSING
3682  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$ERDF
013722  000006                          .WORD   6
013724  005055                          .WORD   EM0
013726  006700                          .WORD   ERR6
013730                          3$:     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
```

```
3739  014022  004000                  8$:     004000                  ;LOAD MAR HI
3740  014024                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3741  014024  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3742  014030  040620                          040620                  ;MOVE MEMORY TO THE BR
3743  014032                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3744  014032  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3745  014036  061225                          61225                   ;MOV BR TO PORT5
3746  014040  010205                          MOV     R2,R5           ;PUT "EXPECTED" IN R5
3747  014042  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" IN R4
3748  014046  120504                          CMPB    R5,R4           ;DATA CORRECT?
3749  014050  001406                          BEQ     6$              ;BR IF YES
3750  014052                                  ERROR   6               ;ADDRESSING ERROR
3751  014056  104455                          TRAP    C$ERDF
3752  014060  000006                          .WORD   6
3753  014062  005055                          .WORD   EM0
3754  014064  006700                          .WORD   ERR6
3755  014066                          6$:     ESCAPE TST
3756  014066  104410                          TRAP    C$ESCAPE
3757  014070  000020                          .WORD   L10052-.
3758  014072                                  BREAK
3759  014072  104422                          TRAP    C$BRK
3760  014074  005237  002406                  INC     FLAG            ;NEXT ADDRESS
3761  014100  023737  002436  002406          CMP     MEMSZ,FLAG      ;IS IT THE LAST
3762  014106  001325                          BNE     4$              ;BR IF NO
3763  014110                          9$:
3764  014110                          ENDTST
3765  014110                          L10052:
3766  014110  104401                          TRAP    C$ETST
3767
3768  014112                                  BADHEAD
3769                                          ;************** TEST 9 **************
3770                                          ;*IOP MAR TEST
3771                                          ;*PERFORM DUAL ADDRESSING TEST
3772                                          ;*USING MAR AUTO-INC FEATURE
3773  014112                                  BADHEAD
3774                                          ;************** TEST 9 **************
3775
3776  014112                          BGNTST
3777  014112                          T9::
3778  014112                                  K4ONLY                  ;FOR 4K CPUS ONLY.
3779                                          ;DO NOT DO TEST IF M8200, OR M8204
3780  014122  104432                          TRAP    C$EXIT
3781  014124  000342                          .WORD   L10053-.
3782  014126                                  MYINT
3783  014126  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
3784                                                                  ;R1 CONTAINS BASE M8200,4,7 ADDRESS
3785  014132                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
3786  014136  005002                          CLR     R2              ;START WITH A ZERO
3787  014140  013703  002436                  MOV     MEMSZ,R3        ;GET MEMORY SIZE
3788  014144  005203                          INC     R3              ;STOP ADDR=MEMSZ+1
3789  014146                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3790  014146  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3791  014152  010000                          010000                  ;LOAD MAR WITH A ZERO
3792  014154                                  CLRMAR
3793  014154  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3794  014162  010261  000004          1$:     MOV     R2,4(R1)        ;WRITE DATA TO PORT4
```

```
3795  014166
3796  014166  004537  003044          ROMCLK                          ;NEXT WORD IS INSTRUCTION,
3797  014172  136500                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
3798  014174  005202                  136500                          ;MEM PORT4, AUTO-INC MAR
3799  014176  020302                  INC     R2                      ;INCREMENT DATA
3800  014200  001370                  CMP     R3,R2   ;DONE YET?
3801  014202  005002                  BNE     1$                      ;BR IF NO
3802  014204                          CLR     R2                      ;RESTART WITH A ZERO
3803  014204  004537  003044          ROMCLK                          ;NEXT WORD IS INSTRUCTION,
3804  014210  010000                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
3805  014212                          010000                          ;LOAD MAR WITH A ZERO
3806  014212  004537  003044          CLRMAR
3807  014220                          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
3808  014220                  2$:
3809  014220  004537  003100          SROMCLK                         ;NEXT WORD IS INSTRUCTION,
3810  014224  055224                  JSR     R5,..SROMCLK
3811  014226  010205                  055224                          ;MOVE MEM TO PORT4
3812  014230  116104  000004          MOV     R2,R5                   ;PUT "EXPECTED" IN R5
3813  014234  120504                  MOVB    4(R1),R4                ;PUT "FOUND" IN R4
3814  014236  001406                  CMPB    R5,R4                   ;DATA CORRECT?
3815  014240                          BEQ     3$                      ;BR IF YES
3816  014244  104455                  ERROR   11                      ;MAR ERROR
3817  014246  000013                  TRAP    C$ERDF
3818  014250  005055                  .WORD   11
3819  014252  007276                  .WORD   EM0
3820  014254                          .WORD   ERR11
3821  014254  004537  003100  3$:     SROMCLK
3822  014260  000000                  JSR     R5,..SROMCLK
3823  014262  005004                  0                               ;DUMP NOP INSTR. TO CLK AUTO INC IN MAR.
3824  014264                          CLR     R4
3825  014264  004537  003044          ROMCLK          ;READ IBUS* <15> (MAR HIGH)
3826  014270  121325                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
3827                                  121325          ;MAR HIGH _POT 5
3828  014272
3829  014272  004537  003044          ROMCLK          ;READ IBUS* <14> (MAR LOW)
3830  014276  121304                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
3831  014300  016104  000004          121304
3832  014304  042704  160000          MOV     4(R1),R4                ;ADD TO MAR HIGH.
3833  014310  005202                  BIC     #160000,R4
3834  014312  020237  002436          INC     R2
3835  014316  001002                  CMP     R2,MEMSZ
3836  014320  052702  010000          BNE     35$
3837  014324                          BIS     #10000,R2               ;IF AT HIGH LIMIT,ADD IN OVERFLOW BIT.
3838  014324  020204          35$:
3839  014326  001406                  CMP     R2,R4   ;ADDR. OK?
3840  014330                          BEQ     4$
3841  014334  104455                  ERROR   11                      ;ERROR MAR ADDR. BAD IN IBUS <14>AND <15>
3842  014336  000013                  TRAP    C$ERDF
3843  014340  005055                  .WORD   11
3844  014342  007276                  .WORD   EM0
3845                                  .WORD   ERR11
3846                                                  ;EXPECTED (R4) IS COMBINATION OF
3847  014344                                          ;IBUS* <14> AND <15>
3848  014344                  4$:
3849  014344  104410                  ESCAPE  TST
3850  014346  000120                  TRAP    C$ESCAPE
                                      .WORD   L10053-.
```

```
3851  014350                                BREAK
3852  014350  104422                        TRAP    C$BRK
3853  014352  032702  010000                BIT     #10000,R2       ;DONE YET?
3854  014356  001720                        BEQ     2$
3855                                        :*
3856                                        :*THIS SECTION OF CODE ADDED TO MAKE SURE
3857                                        :*THAT MASTER CLEAR, CLEARS THE MAR
3858                                        :*
3859
3860  014360                                SKIP06  40$
3861                                        ;GOTO 40$ IF M8206
3862  014370  005737  002470                TST     RUNB
3863  014374  001034                        BNE     40$
3864  014376  005737  002472                TST     RUNINH
3865  014402  001031                        BNE     40$
3866  014404  052711  040000                BIS     #40000,(R1)     ;SET MASTER CLEAR
3867  014410  005011                        CLR     (R1)            ;CLEAR MASTER CLEAR
3868  014412                                ROMCLK                  ;WE MUST FIRST CLOCK
3869  014412  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3870  014416  121325                        121325                  ;THE MAR LATCH REGS
3871  014420                                ROMCLK                  ;BEFORE WE CAN READ THEM
3872  014420  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3873  014424  121304                        121304
3874  014426                                ROMCLK                  ;READ IBUS* <15> PUT IN PORT5
3875  014426  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3876  014432  121325                        121325                  ;MAR HIGH
3877  014434                                ROMCLK                  ;READ IBUS* <14>, PUT IN PORT4
3878  014434  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3879  014440  121304                        121304                  ;MAR LOW
3880  014442  005002                        CLR     R2              ;EXPECT MAR CLEAR
3881  014444  016104  000004                MOV     4(R1),R4        ;READ PORTS 4&5. THEY CONTAIN
3882                                                                ;THE CONTENTS OF THE MAR
3883                                                                ;MASTER CLEAR SHOULD HAVE
3884                                                                ;CLEARED THE MAR
3885                                                                ;BRANCH END TST IF CLEAR
3886  014450  001406                        BEQ     40$
3887  014452                                ERROR   44
3888  014456  104455                        TRAP    C$ERDF
3889  014460  000054                        .WORD   44
3890  014462  005055                        .WORD   EMO
3891  014464  011054                        .WORD   ERR44
3892  014466                        40$:
3893  014466                        ENDTST
3894  014466                        L10053:
3895  014466  104401                        TRAP    C$ETST
3896
3897  014470                                BADHEAD
3898                                        :************** TEST 10 **************
3899                                        :*IOP (CRAM) ODT BITS TEST
3900                                        :*LOAD MAR WITH A 0 INC MAR UNTIL IT OVERFLOWS
3901                                        :*VERIFY THAT IBUS* 10 BITS IS SET ONLY WHEN MAR BIT 8 IS A ONE
3902                                        :*AND THAT IBUS* 10 BIT6 IS SET ON MAR OVERFLOW
3903  014470                                BADHEAD
3904                                        :************** TEST 10 **************
3905
3906  014470                        BGNTST
```

```
3907  014470                                T10::
3908  014470                                      MACEX
3909                                        ;DO NOT DO TEST IF M8200
3910  014476  104432                       TRAP    C$EXIT
3911  014500  000234                       .WORD   L10054-.
3912  014502                                MYINT
3913  014502  013701  002516               MOV     KMCSR,R1            ;RECORD DEVICE ADDR.
3914                                                                    ;R1 CONTAINS BASE M8200,4,7 ADDRESS
3915  014506                                MSTCLR                      ;MASTER CLEAR M820C,4,7
3916  014512  005002                       CLR     R2                  ;R2=SAME AS MAR CONTENTS
3917  014514                                ROMCLK                      ;NEXT WORD IS INSTRUCTION,
3918  014514  004537  003044               JSR     R5,..ROMCLK         ;CLOCK INSTRUCTION
3919  014520  010000                       010000                      ;MAR_0
3920  014522                                1$:
3921  014522                                ROMCLK                      ;NEXT WORD IS INSTRUCTION,
3922  014522  004537  003044               JSR     R5,..ROMCLK         ;CLOCK INSTRUCTION
3923  014526  121204                       121204                      ;PORT4=IBUS*10
3924  014530  005005                       CLR     R5                  ;R5="EXPECTED"
3925  014532  032702  000400               BIT     #BIT8,R2            ;IS BIT8 SET IN MAR?
3926  014536  001402                       BEQ     .+6                 ;BR IF NO
3927  014540  012705  000040               MOV     #BIT5,R5            ;IF YES THEN SET BITS
3928  014544  016104  000004               MOV     4(R1),R4            ;R4="FOUND"
3929  014550  042704  177637               BIC     #177637,R4          ;CLEAR UNWANTED BITS
3930  014554  020504                       CMP     R5,R4               ;BITS 5&6 SHOULD BE CLEAR
3931  014556  001410                       BEQ     15$                 ;BR IF OK
3932  014560                                ERROR   7                   ;ERROR BITS 5&6 NOT CLEAR
3933  014564  104455                       TRAP    C$ERDF
3934  014566  000007                       .WORD   7
3935  014570  005055                       .WORD   EM0
3936  014572  007026                       .WORD   ERR7
3937  014574                                ESCAPE  TST
3938  014574  104410                       TRAP    C$ESCAPE
3939  014576  000136                       .WORD   L10054-.
3940  014600                                15$:
3941  014600                                ROMCLK                      ;NEXT WORD IS INSTRUCTION,
3942  014600  004537  003044               JSR     R5,..ROMCLK         ;CLOCK INSTRUCTION
3943  014604  014000                       014000                      ;INC MAR
3944  014606  005202                       INC     R2                  ;BUMP MEM ADDRESS
3945  014610  022702  002000               CMP     #2000,R2            ;OVERFLOWED YET?(OVFL PAGE BITS).
3946  014614  001342                       BNE     1$                  ;BR IF NO
3947  014616                                ROMCLK                      ;NEXT WORD IS INSTRUCTION,
3948  014616  004537  003044               JSR     R5,..ROMCLK         ;CLOCK INSTRUCTION
3949  014622  121204                       121204                      ;PART4 IBUS* 10
3950  014624  012705  000100               MOV     #BIT6,R5            ;R5="EXPECTED"
3951  014630  016104  000004               MOV     4(R1),R4            ;R4="FOUND"
3952  014634  042704  177627               BIC     #177627,R4          ;CLEAR UNWANTED BITS
3953  014640  020504                       CMP     R5,R4               ;BIT6 SHOULD BE SET
3954  014642  001406                       BEQ     17$                 ;BR IF OK
3955  014644                                ERROR   7                   ;ERROR, BIT6 NOT SET
3956  014650  104455                       TRAP    C$ERDF
3957  014652  000007                       .WORD   7
3958  014654  005055                       .WORD   EM0
3959  014656  007026                       .WORD   ERR7
3960  014660                                17$:    ROMCLK              ;NEXT WORD IS INSTRUCTION,
3961  014660  004537  003044               JSR     R5,..ROMCLK         ;CLOCK INSTRUCTION
3962  014664  010000                       010000                      ;MAR_0
```

```
3963  014666
3964  014666  004537  003044                ROMCLK                          ;NEXT WORD IS INSTRUCTION,
3965  014672  004000                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
3966  014674                                004000                          ;MAR HI 0
3967  014674  004537  003044                ROMCLK                          ;NEXT WORD IS INSTRUCTION,
3968  014700  121204                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
3969  014702  005005                        121204                          ;PORT4 IBUS= 10
3970  014704  016104  000004                CLR     R5                      ;R5="EXPECTED"
3971  014710  042704  177637                MOV     4(R1),R4                ;R4="FOUND"
3972  014714  020504                        BIC     #177637,R4              ;CLEAR UNWANTED BITS
3973  014716  001406                        CMP     R5,R4                   ;BITS 5&6 SHOULD BE CLEAR
3974  014720                                BEQ     2$                      ;BR IF OK
3975  014724  104455                        ERROR   7                       ;ERROR 5&6 NOT BOTH CLEAR
3976  014726  000007                        TRAP    C$ERDF
3977  014730  005055                        .WORD   7
3978  014732  007026                        .WORD   EM0
3979  014734                                .WORD   ERR7
3980  014734                        2$:
3981  014734                        ENDTST
3982  014734  104401                L10054:
3983                                        TRAP    C$ETST
3984  014736                                BADHEAD
3985                                        ;************** TEST 11 **************
3986                                        ;*CRAM TEST OF JUMP(I) NEVER MICRO-PROCESSOR INSTRUCTION.
3987                                        ;*PERFORM THE JUMP INSTRUCTION
3988                                        ;*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
3989                                        ;*IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
3990                                        ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
3991                                        ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
3992                                        ;*THE CRAM PC IS CORRECT. IF THE CRAM PC IS NOT RIGHT,
3993                                        ;*THEN PORT4 CONTAINS A 37
3994  014736                                BADHEAD
3995                                        ;************** TEST 11 **************
3996
3997  014736                        BGNTST
3998  014736                        T11::
3999  014736                                SKIP04  10$
4000                                        ;GOTO 10$ IF M8204
4001  014746                                EXIT    TST                             ;CAN'T DO IF ROM,4K
4002  014746  104432                        TRAP    C$EXIT
4003  014750  000230                        .WORD   L10055-.
4004  014752                        10$:
4005  014752                                MYINT
4006  014752  013701  002516                MOV     KMCSR,R1                        ;RECORD DEVICE ADDR.
4007                                                                        ;R1 CONTAINS BASE M8200,4,7 ADDRESS
4008  014756                                MSTCLR                          ;MASTER CLEAR M8200,4,7
4009  014762                                BGNSEG
4010  014762  104404                        TRAP    C$BSEG
4011  014764  004737  003474                JSR     PC,MEMSET               ;SET MEM AND RAM
4012  014770                        1$:
4013  014770  004737  003166                JSR     PC,CLRALL               ;CLEAR ALL CONDITIONS
4014  014774                                SROMCLK                         ;NEXT WORD IS INSTRUCTION,
4015  014774  004537  003100                JSR     R5,..SROMCLK
4016  015000  100400                        100400                          ;START AT ROM PC=0
4017  015002                                SROMCLK                         ;NEXT WORD IS INSTRUCTION,
4018  015002  004537  003100                JSR     R5,..SROMCLK
```

```
4019  015006  114377                        114377!<400*0>          ;JUMP TO ROM PC OF 1777
4020  015010  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4021  015014  000001                        1                       ;EXPECTED DATA
4022  015016  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
4023  015020  001406                        BEQ     2$              ;BR IF NO
4024  015022                                ERROR   5               ;ERROR, CRAM PC IS WRONG
4025  015026  104455                        TRAP    C$ERDF
4026  015030  000005                        .WORD   5
4027  015032  005055                        .WORD   EM0
4028  015034  006556                        .WORD   ERR5
4029  015036                          2$:   ESCAPE SEG
4030  015036  104410                        TRAP    C$ESCAPE
4031  015040  000002                        .WORD   10000$-.
4032  015042                                ENDSEG
4033  015042                          10000$:
4034  015042  104405                        TRAP    C$ESEG
4035  015044                                BGNSEG
4036  015044  104404                        TRAP    C$BSEG
4037  015046  004737  003166                JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
4038  015052                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4039  015052  004537  003100                JSR     R5,..SROMCLK
4040  015056  100403                        100403                  ;START AT ROM PC=3
4041  015060                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4042  015060  004537  003100                JSR     R5,..SROMCLK
4043  015064  100000                        100000!<400*0>  ;JUMP TO ROM PC OF 0
4044  015066  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4045  015072  000004                        4                       ;EXPECTED DATA
4046  015074  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
4047  015076  001406                        BEQ     4$              ;BR IF YES
4048  015100                                ERROR   5               ;ERROR, CROM PC IS WRONG
4049  015104  104455                        TRAP    C$ERDF
4050  015106  000005                        .WORD   5
4051  015110  005055                        .WORD   EM0
4052  015112  006556                        .WORD   ERR5
4053  015114                          4$:   ESCAPE SEG
4054  015114  104410                        TRAP    C$ESCAPE
4055  015116  000002                        .WORD   10001$-.
4056  015120                                ENDSEG
4057  015120                          10001$:
4058  015120  104405                        TRAP    C$ESEG
4059  015122                                BGNSEG
4060  015122  104404                        TRAP    C$BSEG
4061  015124  004737  003166                JSR     PC,CLRALL       ;CLEAR ALL CONDITINS
4062  015130                                SROMCLK                 ;NEXT WORD IS INSTRUCION,
4063  015130  004537  003100                JSR     R5,..SROMCLK
4064  015134  100406                        100406                  ;START AT ROM PC=6
4065  015136                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4066  015136  004537  003100                JSR     R5,..SROMCLK
4067  015142  104125                        104125:<400*0>          ;JUMP TO ROM PC OF 525
4068  015144  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4069  015150  000007                        7                       ;EXPECTED DATA
4070  015152  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
4071  015154  001406                        BEQ     6$              ;BR IF YES
4072  015156                                ERROR   5               ;ERROR, CRAM PC IS WRONG
4073  015162  104455                        TRAP    C$ERDF
4074  015164  000005                        .WORD   5
```

```
4075  015166  005055                        .WORD   EM0
4076  015170  006556                        .WORD   ERR5
4077  015172                        6$:     ESCAPE SEG
4078  015172  104410                        TRAP    C$ESCAPE
4079  015174  000002                        .WORD   10002$-.
4080  015176                                ENDSEG
4081  015176                        10002$:
4082  015176  104405                        TRAP    C$ESEG
4083  015200                        ENDTST
4084  015200                        L10055:
4085  015200  104401                        TRAP    C$ETST
4086
4087  015202                                BADHEAD
4088                                        ;************** TEST 12 **************
4089                                        ;*CRAM TEST OF JUMP(I) ALWAYS MICRO-PROCESSOR INSTRUCTION.
4090                                        ;*PERFORM THE JUMP INSTRUCTION
4091                                        ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
4092                                        ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4093                                        ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4094                                        ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4095                                        ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
4096                                        ;*THEN PORT4 WILL CONTAIN A 37
4097  015202                                BADHEAD
4098                                        ;************** TEST 12 **************
4099
4100  015202                        BGNTST
4101  015202                        T12::
4102  015202                                MACEX2                  ;DON'T DO IF M8200
4103                                        ;DO NOT DO TEST IF M8200
4104  015212  104432                        TRAP    C$EXIT
4105  015214  000214                        .WORD   L10056-.
4106  015216                                MYINT
4107  015216  013701  002516                MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
4108                                                                ;R1 CONTAINS BASE M8200,4,7 ADDRESS
4109  015222                                MSTCLR                  ;MASTER CLEAR M8200,4,7
4110  015226  004737  003474                JSR     PC,MEMSET       ;SET MEM AND RAM
4111  015232                        1$:     BGNSEG
4112  015232  104404                        TRAP    C$BSEG
4113  015234                                SROMCLK                 ;NEXT WORD IS INSTRUCTION.
4114  015234  004537  003100                JSR     R5,.SROMCLK
4115  015240  100400                        100400                  ;START AT ROM PC=0
4116  015242                                SROMCLK                 ;NEXT WORD IS INSTRUCION.
4117  015242  004537  003100                JSR     R5,.SROMCLK
4118  015246  114777                        114377!<400*1>          ;JUMP TO ROM PC OF 1777
4119  015250  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4120  015254  000377                        377                     ;EXPECTED DATA
4121  015256  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
4122  015260  001406                        BEQ     2$              ;BR IF YES
4123  015262                                ERROR   5               ;ERROR, CRAM PC IS WRONG
4124  015266  104455                        TRAP    C$ERDF
4125  015270  000005                        .WORD   5
4126  015272  005055                        .WORD   EM0
4127  015274  006556                        .WORD   ERR5
4128  015276                        2$:     ESCAPE SEG
4129  015276  104410                        TRAP    C$ESCAPE
4130  015300  000002                        .WORD   10000$-.
```

```
4131  015302                                    ENDSEG
4132  015302                              10000$:
4133  015302  104405                            TRAP    C$ESEG
4134  015304                                    BGNSEG
4135  015304  104404                            TRAP    C$BSEG
4136  015306                                    SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4137  015306  004537  003100                    JSR     R5,.SROMCLK
4138  015312  100403                            100403                  ;START AT ROM PC=3
4139  015314                                    SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4140  015314  004537  003100                    JSR     R5,.SROMCLK
*4141 015320  100400                            100000!<400*1>  ;JUMP TO ROM PC OF 0
4142  015322  004737  003330                    JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4143  015326  000000                            0                       ;EXPECTED DATA
4144  015330  120504                            CMPB    R5,R4           ;IS ROM PC CORRECT?
4145  015332  001406                            BEQ     4$              ;BR IF YES
4146  015334                                    ERROR   5       ;ERROR, CRAM PC IS WRONG
4147  015340  104455                            TRAP    C$ERDF
4148  015342  000005                            .WORD   5
4149  015344  005055                            .WORD   EM0
4150  015346  006556                            .WORD   ERR5
4151  015350                              4$:   ESCAPE SEG
4152  015350  104410                            TRAP    C$ESCAPE
4153  015352  000002                            .WORD   10001$-.
4154  015354                                    ENDSEG
4155  015354                              10001$:
4156  015354  104405                            TRAP    C$ESEG
4157  015356                                    BGNSEG
4158  015356  104404                            TRAP    C$BSEG
4159  015360                                    SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4160  015360  004537  003100                    JSR     R5,.SROMCLK
4161  015364  100406                            100406                  ;START AT ROM PC=6
4162  015366                                    SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4163  015366  004537  003100                    JSR     R5,.SROMCLK
4164  015372  104525                            104125!<400*1>          ;JUMP TO ROM PC OF 525
4165  015374  004737  003330                    JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4166  015400  000125                            125                     ;EXPECTED DATA
4167  015402  120504                            CMPB    R5,R4           ;IS ROM PC CORRECT?
4168  015404  001406                            BEQ     6$              ;BR IF YES
4169  015406                                    ERROR   5               ;ERROR, CRAM PC IS WRONG
4170  015412  104455                            TRAP    C$ERDF
4171  015414  000005                            .WORD   5
4172  015416  005055                            .WORD   EM0
4173  015420  006556                            .WORD   ERR5
4174  015422                              6$:   ESCAPE SEG
4175  015422  104410                            TRAP    C$ESCAPE
4176  015424  000002                            .WORD   10002$-.
4177  015426                                    ENDSEG
4178  015426                              10002$:
4179  015426  104405                            TRAP    C$ESEG
4180  015430                              ENDTST
4181  015430                              L10056:
4182  015430  104401                            TRAP    C$ETST
4183
4184  015432                                    BADHEAD
4185                                            ;************** TEST 13 ***************
4186                                            ;*CRAM TEST OF JUMP(1) ON C BIT SET MICRO-PROCESSOR INSTRUCTION.
```

```
 4187
 4188
 4189
 4190
 4191
 4192
 4193
 4194  015432                                  ;*SET THE C BIT, PERFORM THE JUMP INSTRUCTION.
 4195                                          ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
 4196                                          ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
 4197  015432                                  ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
 4198  015432                                  ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
 4199  015432                                  ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
 4200                                          ;*THEN PORT4 WILL CONTAIN A 37
 4201  015442  104432                         BADHEAD
 4202  015444  000230                         ;************** TEST 13 **************
 4203  015446
 4204  015446  013701  002516          BGNTST
 4205                                   T13::
 4206  015452                           MACEX2                          ;DON'T DO IF M8200
 4207  015456  004737  003474                  ;DO NOT DO TEST IF M8200
 4208  015462                                  TRAP    C$EXIT
 4209  015462  104404                         .WORD   L10057-.
 4210  015464  004737  003260          MYINT
 4211  015470                           MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
 4212  015470  004537  003100                                          ;R1 CONTAINS BASE M8200,4,7 ADDRESS
 4213  015474  100400                  MSTCLR                          ;MASTER CLEAR M8200,4,7
 4214  015476                           JSR     PC,MEMSET               ;SET MEM AND RAM
 4215  015476  004537  003100  1$:     BGNSEG
 4216  015502  115377                  TRAP    C$BSEG
 4217  015504  004737  003330          JSR     PC,SETC                 ;SET THE C BIT'
 4218  015510  000377                  SROMCLK                         ;NEXT WORD IS INSTRUCTION,
 4219  015512  120504                  JSR     R5,.SROMCLK
 4220  015514  001406                  100400                          ;START AT ROM PC=0
 4221  015516                           SROMCLK                         ;NEXT WORD IS INSTRUCION,
 4222  015522  104455                  JSR     R5,.SROMCLK
 4223  015524  000005                  114377!<400*2>                  ;JUMP TO ROM PC OF 1777
 4224  015526  005055                  JSR     PC,RAMDAT               ;R4=CRAM PC (LSB 8 BITS)
 4225  015530  006556                  377                             ;EXPECTED DATA
 4226  015532                           CMPB    R5,R4                   ;IS ROM PC CORRECT?
 4227  015532                           BEQ     2$                      ;BR IF YES
 4228  015532  104410                  ERROR   5                       ;ERROR, CRAM PC IS WRONG
 4229  015534  000002                  TRAP    C$ERDF
 4230  015536                           .WORD   5
 4231  015536                           .WORD   EM0
 4232  015536  104405                  .WORD   ERR5
 4233  015540                   2$:
 4234  015540  104404                  ESCAPE SEG                      ;LOOP TO 1$ IF SW09=1
 4235  015542  004737  003260          TRAP    C$ESCAPE
 4236  015546                           .WORD   10000$-.
 4237  015546  004537  003100          ENDSEG
 4238  015552  100403           10000$:
 4239  015554                           TRAP    C$ESEG
 4240  015554  004537  003100          BGNSEG
 4241  015560  101000                  TRAP    C$BSEG
 4242  015562  004737  003330          JSR     PC,SETC                 ;SET THE C BIT'
                                       SROMCLK                         ;NEXT WORD IS INSTRUCTION,
                                       JSR     R5,.SROMCLK
                                       100403                          ;START AT ROM PC=3
                                       SROMCLK                         ;NEXT WORD IS INSTRUCTION,
                                       JSR     R5,.SROMCLK
                                       100000!<400*2>                  ;JUMP TO ROM PC OF 0
                                       JSR     PC,RAMDAT               ;R4-CRAM PC (LSB 8 BITS)
```

```
4243  015566  000000                          0                       ;EXPECTED DATA
4244  015570  120504                          CMPB    R5,R4           ;IS ROM PC CORRECT?
4245  015572  001406                          BEQ     4$              ;BR IF YES
4246  015574                                  ERROR   5               ;ERROR, CRAM PC IS WRONG
4247  015600  104455                          TRAP    C$ERDF
4248  015602  000005                          .WORD   5
4249  015604  005055                          .WORD   EM0
4250  015606  006556                          .WORD   ERR5
4251  015610                          4$:
4252  015610                                  ESCAPE SEG              ;LOOP TO 3$ IF SW09=1
4253  015610  104410                          TRAP    C$ESCAPE
4254  015612  000002                          .WORD   10001$-.
4255  015614                                  ENDSEG
4256  015614                          10001$:
4257  015614  104405                          TRAP    C$ESEG
4258  015616                                  BGNSEG
4259  015616  104404                          TRAP    C$BSEG
4260  015620  004737  003260                  JSR     PC,SETC         ;SET THE C BIT'
4261  015624                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4262  015624  004537  003100                  JSR     R5,.SROMCLK
4263  015630  100406                          100406                  ;START AT ROM PC=6
4264  015632                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4265  015632  004537  003100                  JSR     R5,.SROMCLK
4266  015636  105125                          104125!<400*2>          ;JUMP TO ROM PC OF 525
4267  015640  004737  003330                  JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4268  015644  000125                          125                     ;EXPECTED DATA
4269  015646  120504                          CMPB    R5,R4           ;IS ROM PC CORRECT?
4270  015650  001406                          BEQ     6$              ;BR IF YES
4271  015652                                  ERROR   5               ;ERROR, CRAM PC IS WRONG
4272  015656  104455                          TRAP    C$ERDF
4273  015660  000005                          .WORD   5
4274  015662  005055                          .WORD   EM0
4275  015664  006556                          .WORD   ERR5
4276  015666                          6$:     ESCAPE SEG
4277  015666  104410                          TRAP    C$ESCAPE
4278  015670  000002                          .WORD   10002$-.
4279  015672                                  ENDSEG
4280  015672                          10002$:
4281  015672  104405                          TRAP    C$ESEG
4282  015674                          ENDTST
4283  015674                          L10057:
4284  015674  1044^1                          TRAP    C$ETST
4285
4286  015676                                  BADHEAD
4287                                          ;************** TEST 14 ***************
4288                                          ;*CRAM TEST OF JUMP(I) ON Z BIT SET MICRO-PROCESSOR INSTRUCTION.
4289                                          ;*SET THE Z BIT, PERFORM THE JUMP INSTRUCTION.
4290                                          ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
4291                                          ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4292                                          ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4293                                          ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4294                                          ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
4295                                          ;*THEN PORT4 WILL CONTAIN A 37
4296  015676                                  BADHEAD
4297                                          ;************** TEST 14 **************
4298
```

```
4299  015676                                BGNTST
4300  015676                                T14::
4301  015676                                      MACEX2                    ;DON'T DO IF M8200.
4302                                              ;DO NOT DO TEST IF M8200
4303  015706  104432                              TRAP    C$EXIT
4304  015710  000230                              .WORD   L10060-.
4305  015712                                      MYINT
4306  015712  013701  002516                      MOV     KMCSR,R1                  ;RECORD DEVICE ADDR.
4307                                                                        ;R1 CONTAINS BASE M8200,4,7 ADDRESS
4308  015716                                      MSTCLR                    ;MASTER CLEAR M8200,4,7
4309  015722  004737  003474                      JSR     PC,MEMSET         ;SET MEM AND RAM
4310  015726                                1$:   BGNSEG
4311  015726  104404                              TRAP    C$BSEG
4312  015730  004737  003312                      JSR     PC,SETZ           ;SET THE Z BIT'
4313  015734                                      SROMCLK                   ;NEXT WORD IS INSTRUCTION,
4314  015734  004537  003100                      JSR     R5,..SROMCLK
4315  015740  100400                              100400                    ;START AT ROM PC=0
4316  015742                                      SROMCLK                   ;NEXT WORD IS INSTRUCION,
4317  015742  004537  003100                      JSR     R5,..SROMCLK
4318  015746  115777                              114377!<400*3>            ;JUMP TO ROM PC OF 1777
4319  015750  004737  003330                      JSR     PC,RAMDAT         ;R4=CRAM PC (LSB 8 BITS)
4320  015754  000377                              377                       ;EXPECTED DATA
4321  015756  120504                              CMPB    R5,R4             ;IS ROM PC CORRECT?
4322  015760  001406                              BEQ     2$                ;BR IF YES
4323  015762                                      ERROR   5                 ;ERROR, CRAM PC IS WRONG
4324  015766  104455                              TRAP    C$ERDF
4325  015770  000005                              .WORD   5
4326  015772  005055                              .WORD   EM0
4327  015774  006556                              .WORD   ERR5
4328  015776                                2$:   ESCAPE SEG
4329  015776  104410                              TRAP    C$ESCAPE
4330  016000  000002                              .WORD   10000$-.
4331  016002                                      ENDSEG
4332  016002                                10000$:
4333  016002  104405                              TRAP    C$ESEG
4334  016004                                      BGNSEG
4335  016004  104404                              TRAP    C$BSEG
4336  016006  004737  003312                      JSR     PC,SETZ           ;SET THE Z BIT'
4337  016012                                      SROMCLK                   ;NEXT WORD IS INSTRUCTION,
4338  016012  004537  003100                      JSR     R5,..SROMCLK
4339  016016  100403                              100403                    ;START AT ROM PC=3
4340  016020                                      SROMCLK                   ;NEXT WORD IS INSTRUCTION,
4341  016020  004537  003100                      JSR     R5,..SROMCLK
4342  016024  101400                              100000!<400*3>            ;JUMP TO ROM PC OF 0
4343  016026  004737  003330                      JSR     PC,RAMDAT         ;R4=CRAM PC (LSB 8 BITS)
4344  016032  000000                              0                         ;EXPECTED DATA
4345  016034  120504                              CMPB    R5,R4             ;IS ROM PC CORRECT?
4346  016036  001406                              BEQ     4$                ;BR IF YES
4347  016040                                      ERROR   5                 ;ERROR, CRAM PC IS WRONG
4348  016044  104455                              TRAP    C$ERDF
4349  016046  000005                              .WORD   5
4350  016050  005055                              .WORD   EM0
4351  016052  006556                              .WORD   ERR5
4352  016054                                4$:   ESCAPE SEG
4353  016054  104410                              TRAP    C$ESCAPE
4354  016056  000002                              .WORD   10001$-.
```

```
                                        ENDSEG
                                10001$:
016060                                  TRAP    C$ESEG
016060                                  BGNSEG
016060  104405                          TRAP    C$BSEG
016062                                  JSR     PC,SETZ         ;SET THE Z BIT'
016062  104404                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
016064  004737  003312                  JSR     R5,.SROMCLK
016070                                  100406                  ;START AT ROM PC=6
016070  004537  003100                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
016074  100406                          JSR     R5,.SROMCLK
016076                                  104125!<400*3>          ;JUMP TO ROM PC OF 525
016076  004537  003100                  JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
016102  105525                          125                     ;EXPECTED DATA
016104  004737  003330                  CMPB    R5,R4           ;IS ROM PC CORRECT?
016110  000125                          BEQ     6$              ;BR IF YES
016112  120504                          ERROR   5               ;ERROR, CRAM PC IS WRONG
016114  001406                          TRAP    C$ERDF
016116                                  .WORD   5
016122  104455                          .WORD   EM0
016124  000005                          .WORD   ERR5
016126  005055                  6$:     ESCAPE SEG
016130  006556                          TRAP    C$ESCAPE
016132                                  .WORD   10002$-.
016132  104410                          ENDSEG
016134  000002                  10002$:
016136                                  TRAP    C$ESEG
016136                          ENDTST
016136  104405                  L10060:
016140                                  TRAP    C$ETST
016140
016140  104401
4385
016142                                  BADHEAD
4387                                    ;************** TEST 15 **************
4388                                    ;*CRAM TEST OF JUMP(1) ON BR0 SET MICRO-PROCESSOR INSTRUCTION.
4389                                    ;*SET THE BR0 BIT, PERFORM THE JUMP INSTRUCTION.
4390                                    ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
4391                                    ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4392                                    ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4393                                    ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4394                                    ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
4395                                    ;*THEN PORT4 WILL CONTAIN A 37
016142                                  BADHEAD
4397                                    ;************** TEST 15 **************
4398
016142                          BGNTST
016142                          T15::
016142                                  MACEX2                  ;DON'T DO IF M8200.
4402                                    ;DO NOT DO TEST IF M8200
016152  104432                          TRAP    C$EXIT
016154  000230                          .WORD   L10061-.
016156                                  MYINT
016156  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
4407                                                            ;R1 CONTAINS BASE M8200.4.7 ADDRESS
016162                                  MSTCLR                  ;MASTER CLEAR M8200.4.7
016166  004737  003474                  JSR     PC,MEMSET       ;SET MEM AND RAM
016172                          1$:
```

```
4411  016172
4412  016172  104404
4413  016174  004737  003220
4414  016200
4415  016200  004537  003100
4416  016204  100400
4417  016206
4418  016206  004537  003100
4419  016212  116377
4420  016214  004737  003330
4421  016220  000377
4422  016222  120504
4423  016224  001406
4424  016226
4425  016232  104455
4426  016234  000005
4427  016236  005055
4428  016240  006556
4429  016242
4430  016242  104410
4431  016244  000002
4432  016246
4433  016246
4434  016246  104405
4435  016250
4436  016250  104404
4437  016252  004737  003220
4438  016256
4439  016256  004537  003100
4440  016262  100403
4441  016264
4442  016264  004537  003100
4443  016270  102000
4444  016272  004737  003330
4445  016276  000000
4446  016300  120504
4447  016302  001406
4448  016304
4449  016310  104455
4450  016312  000005
4451  016314  005055
4452  016316  006556
4453  016320
4454  016320  104410
4455  016322  000002
4456  016324
4457  016324
4458  016324  104405
4459  016326
4460  016326  104404
4461  016330  004737  003220
4462  016334
4463  016334  004537  003100
4464  016340  100406
4465  016342
4466  016342  004537  003100

                BGNSEG
                TRAP    C$BSEG
                JSR     PC,SETBRO       ;SET THE BRO BIT'
                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
                JSR     R5, SROMCLK
                100400                  ;START AT ROM PC=0
                SROMCLK                 ;NEXT WORD IS INSTRUCION,
                JSR     R5,.SROMCLK
                114377!<400*4>          ;JUMP TO ROM PC OF 1777
                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
                377                     ;EXPECTED DATA
                CMPB    R5,R4           ;IS ROM PC CORRECT?
                BEQ     2$              ;BR IF YES
                ERROR   5               ;ERROR, CRAM PC IS WRONG
                TRAP    C$ERDF
                .WORD   5
                .WORD   EM0
                .WORD   ERR5
2$:             ESCAPE SEG
                TRAP    C$ESCAPE
                .WORD   10000$-.
                ENDSEG
10000$:
                TRAP    C$ESEG
                BGNSEG
                TRAP    C$BSEG
                JSR     PC,SETBRO               ;SET THE BRO BIT'
                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
                JSR     R5,.SROMCLK
                100403                  ;START AT ROM PC=3
                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
                JSR     R5,.SROMCLK
                100000!<400*4>          ;JUMP TO ROM PC OF 0
                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
                0                       ;EXPECTED DATA
                CMPB    R5,R4           ;IS ROM PC CORRECT?
                BEQ     4$              ;BR IF YES
                ERROR   5               ;ERROR, CRAM PC IS WRONG
                TRAP    C$ERDF
                .WORD   5
                .WORD   EM0
                .WORD   ERR5
4$:             ESCAPE SEG
                TRAP    C$ESCAPE
                .WORD   10001$-.
                ENDSEG
10001$:
                TRAP    C$ESEG
                BGNSEG
                TRAP    C$BSEG
                JSR     PC,SETBRO               ;SET THE BRO BIT'
                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
                JSR     R5,.SROMCLK
                100406                  ;START AT ROM PC=6
                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
                JSR     R5,.SROMCLK
```

```
 4467  016346  106125                      104125!<400*4>          ;JUMP TO ROM PC OF 525
 4468  016350  004737  003330              JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
 4469  016354  000125                      125                     ;EXPECTED DATA
 4470  016356  120504                      CMPB    R5,R4           ;IS ROM PC CORRECT?
 4471  016360  001406                      BEQ     6$              ;BR IF YES
 4472  016362                              ERROR   5               ;ERROR, CRAM PC IS WRONG
 4473  016366  104455                      TRAP    C$ERDF
 4474  016370  000005                      .WORD   5
 4475  016372  005055                      .WORD   EMO
 4476  016374  006556                      .WORD   ERR5
 4477  016376                      6$:     ESCAPE SEG
 4478  016376  104410                      TRAP    C$ESCAPE
 4479  016400  000002                      .WORD   10002$-.
 4480  016402                              ENDSEG
 4481  016402                      10002$:
 4482  016402  104405                      TRAP    C$ESEG
 4483  016404                      ENDTST
 4484  016404                      L10061:
 4485  016404  104401                      TRAP    C$ETST
 4486
 4487  016406                              BADHEAD
 4488                                      ;************** TEST 16 **************
 4489                                      ;*CRAM TEST OF JUMP(I) ON BR1 SET MICRO-PROCESSOR INSTRUCTION.
 4490                                      ;*SET THE BR1 BIT, PERFORM THE JUMP INSTRUCTION.
 4491                                      ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
 4492                                      ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
 4493                                      ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
 4494                                      ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
 4495                                      ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
 4496                                      ;*THEN PORT4 WILL CONTAIN A 37
 4497  016406                              BADHEAD
 4498                                      ;************** TEST 16 **************
 4499
 4500  016406                      BGNTST
 4501  016406                      T16::
 4502  016406                              MACEX2                  ;DON'T DO IF M8200.
 4503                                      ;DO NOT DO TEST IF M8200
 4504  016416  104432                      TRAP    C$EXIT
 4505  016420  000230                      .WORD   L10062-.
 4506  016422                              MYINT
 4507  016422  013701  002516              MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
 4508                                                                  ;R1 CONTAINS BASE M8200,4,7 ADDRESS
 4509  016426                              MSTCLR                      ;MASTER CLEAR M8200,4,7
 4510  016432  004737  003474              JSR     PC,MEMSET           ;SET MEM AND RAM
 4511  016436                      1$:
 4512  016436                              BGNSEG
 4513  016436  104404                      TRAP    C$BSEG
 4514  016440  004737  003230              JSR     PC,SETBR1           ;SET THE BR1 BIT'
 4515  016444                              SROMCLK                     ;NEXT WORD IS INSTRUCTION,
 4516  016444  004537  003100              JSR     R5,.SROMCLK
 4517  016450  100400                      100400                      ;START AT ROM PC=0
 4518  016452                              SROMCLK                     ;NEXT WORD IS INSTRUCION,
 4519  016452  004537  003100              JSR     R5,.SROMCLK
 4520  016456  116777                      114377!<400*5>              ;JUMP TO ROM PC OF 1777
 4521  016460  004737  003330              JSR     PC,RAMDAT           ;R4=CRAM PC (LSB 8 BITS)
 4522  016464  000377                      377                         ;EXPECTED DATA
```

```
4523  016466  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
4524  016470  001406                        BEQ     2$              ;BR IF YES
4525  016472                                ERROR   5               ;ERROR, CRAM PC IS WRONG
4526  016476  104455                        TRAP    C$ERDF
4527  016500  000005                        .WORD   5
4528  016502  005055                        .WORD   EM0
4529  016504  006556                        .WORD   ERR5
4530  016506                        2$:     ESCAPE SEG
4531  016506  104410                        TRAP    C$ESCAPE
4532  016510  000002                        .WORD   10000$-.
4533  016512                                ENDSEG
4534  016512                        10000$:
4535  016512  104405                        TRAP    C$ESEG
4536  016514                                BGNSEG
4537  016514  104404                        TRAP    C$BSEG
4538  016516  004737  003230                JSR     PC,SETBR1               ;SET THE BR1 BIT'
4539  016522                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4540  016522  004537  003100                JSR     R5,.SROMCLK
4541  016526  100403                        100403                  ;START AT ROM PC=3
4542  016530                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4543  016530  004537  003100                JSR     R5,.SROMCLK
4544  016534  100400                        100000!<400*5>          ;JUMP TO ROM PC OF 0
4545  016536  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4546  016542  000000                        0                       ;EXPECTED DATA
4547  016544  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
4548  016546  001406                        BEQ     4$              ;BR IF YES
4549  016550                                ERROR   5               ;ERROR, CRAM PC IS WRONG
4550  016554  104455                        TRAP    C$ERDF
4551  016556  000005                        .WORD   5
4552  016560  005055                        .WORD   EM0
4553  016562  006556                        .WORD   ERR5
4554  016564                        4$:     ESCAPE SEG
4555  016564  104410                        TRAP    C$ESCAPE
4556  016566  000002                        .WORD   10001$-.
4557  016570                                ENDSEG
4558  016570                        10001$:
4559  016570  104405                        TRAP    C$ESEG
4560  016572                                BGNSEG
4561  016572  104404                        TRAP    C$BSEG
4562  016574  004737  003230                JSR     PC,SETBR1               ;SET THE BR1 BIT'
4563  016600                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4564  016600  004537  003100                JSR     R5,.SROMCLK
4565  016604  100406                        100406                  ;START AT ROM PC=6
4566  016606                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4567  016606  004537  003100                JSR     R5,.SROMCLK
4568  016612  106525                        104125!<400*5>          ;JUMP TO ROM PC OF 525
4569  016614  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4570  016620  000125                        125                     ;EXPECTED DATA
4571  016622  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
4572  016624  001406                        BEQ     6$              ;BR IF YES
4573  016626                                ERROR   5               ;ERROR, CRAM PC IS WRONG
4574  016632  104455                        TRAP    C$ERDF
4575  016634  000005                        .WORD   5
4576  016636  005055                        .WORD   EM0
4577  016640  006556                        .WORD   ERR5
4578  016642                        6$:     ESCAPE SEG
```

```
4579  016642  104410                    TRAP    C$ESCAPE
4580  016644  000002                    .WORD   10002$-.
4581  016646                            ENDSEG
4582  016646                    10002$:
4583  016646  104405                    TRAP    C$ESEG
4584  016650                    ENDTST
4585  016650                    L10062:
4586  016650  104401                    TRAP    C$ETST
4587
4588  016652                            BADHEAD
4589                                    ;************** TEST 17 **************
4590                                    ;*CRAM TEST OF JUMP(I) ON BR4 SET MICRO-PROCESSOR INSTRUCTION.
4591                                    ;*SET THE BR4 BIT, PERFORM THE JUMP INSTRUCTIUN.
4592                                    ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
4593                                    ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4594                                    ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4595                                    ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4596                                    ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
4597                                    ;*THEN PORT4 WILL CONTAIN A 37
4598  016652                            BADHEAD
4599                                    ;************** TEST 17 **************
4600
4601  016652                    BGNTST
4602  016652                    T17::
4603  016652                            MACEX2                  ;DON'T DO IF M8200.
4604                                    ;DO NOT DO TEST IF M8200
4605  016662  104432                    TRAP    C$EXIT
4606  016664  000230                    .WORD   L10063-.
4607  016666                            MYINT
4608  016666  013701  002516            MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
4609  016672                            MSTCLR                  ;MASTER CLEAR M8200,4,7
4610  016676  004737  003474            JSR     PC,MEMSET       ;SET MEM AND RAM
4611  016702                    1$:
4612  016702                            BGNSEG
4613  016702  104404                    TRAP    C$BSEG
```

```
 4614  016704  004737  003240                  JSR     PC,SETBR4       ;SET THE BR4 BIT'
 4615  016710                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 4616  016710  004537  003100                  JSR     R5,..SROMCLK
 4617  016714  100400                          100400                  ;START AT ROM PC=0
 4618  016716                                  SROMCLK                 ;NEXT WORD IS INSTRUCION,
 4619  016716  004537  003100                  JSR     R5,..SROMCLK
```

```
 4620  016722  117377                    114377:<400+6>          ;JUMP TO ROM PC OF 1777
 4621  016724  004737  003330            JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
```

```
 4622  016730  000377                    377                     ;EXPECTED DATA
 4623  016732  120504                    CMPB    R5,R4           ;IS ROM PC CORRECT?
```

```
  4624  016734  001406                        BEQ     2$              ;BR IF YES
```

```
 4625  016736
 4626  016742  104455          ERROR   5               ;ERROR, CRAM PC IS WRONG
 4627  016744  000005          TRAP    C$ERDF
 4628  016746  005055          .WORD   5
 4629  016750  006556          .WORD   EM0
                               .WORD   ERR5
```

```
4630  016752                          2$:     ESCAPE SEG
4631  016752  104410                          TRAP    C$ESCAPE
4632  016754  000002                          .WORD   10000$-.
4633  016756                                  ENDSEG
4634  016756                          10000$:
4635  016756  104405                          TRAP    C$ESEG
4636  016760                                  BGNSEG
4637  016760  104404                          TRAP    C$BSEG
4638  016762  004737  003240                  JSR     PC,SETBR4               ;SET THE BR4 BIT'
4639  016766                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4640  016766  004537  003100                  JSR     R5,.SROMCLK
4641  016772  100403                          100403                  ;START AT ROM PC=3
4642  016774                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4643  016774  004537  003100                  JSR     R5,.SROMCLK
4644  017000  103000                          100000!<400*6>          ;JUMP TO ROM PC OF 0
4645  017002  004737  003330                  JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4646  017006  000000                          0                       ;EXPECTED DATA
4647  017010  120504                          CMPB    R5,R4           ;IS ROM PC CORRECT?
4648  017012  001406                          BEQ     4$              ;BR IF YES
4649  017014                                  ERROR   5               ;ERROR, CRAM PC IS WRONG
4650  017020  104455                          TRAP    C$ERDF
4651  017022  000005                          .WORD   5
4652  017024  005055                          .WORD   EM0
4653  017026  006556                          .WORD   ERR5
4654  017030                          4$:     ESCAPE SEG
4655  017030  104410                          TRAP    C$ESCAPE
4656  017032  000002                          .WORD   10001$-.
4657  017034                                  ENDSEG
4658  017034                          10001$:
4659  017034  104405                          TRAP    C$ESEG
4660  017036                                  BGNSEG
4661  017036  104404                          TRAP    C$BSEG
4662  017040  004737  003240                  JSR     PC,SETBR4               ;SET THE BR4 BIT'
4663  017044                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4664  017044  004537  003100                  JSR     R5,.SROMCLK
4665  017050  100406                          100406                  ;START AT ROM PC=6
4666  017052                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4667  017052  004537  003100                  JSR     R5,.SROMCLK
4668  017056  107125                          104125!<400*6>          ;JUMP TO ROM PC OF 525
4669  017060  004737  003330                  JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4670  017064  000125                          125                     ;EXPECTED DATA
4671  017066  120504                          CMPB    R5,R4           ;IS ROM PC CORRECT?
4672  017070  001406                          BEQ     6$              ;BR IF YES
4673  017072                                  ERROR   5               ;ERROR, CRAM PC IS WRONG
4674  017076  104455                          TRAP    C$ERDF
4675  017100  000005                          .WORD   5
4676  017102  005055                          .WORD   EM0
4677  017104  006556                          .WORD   ERR5
4678  017106                          6$:     ESCAPE SEG
4679  017106  104410                          TRAP    C$ESCAPE
4680  017110  000002                          .WORD   10002$-.
4681  017112                                  ENDSEG
4682  017112                          10002$:
4683  017112  104405                          TRAP    C$ESEG
4684  017114                          ENDTST
4685  017114                          L10063:
```

```
 4686  017114  104401                          TRAP    C$ETST
 4687
 4688  017116                                  BADHEAD
 4689                                          ;************** TEST 18 ***************
 4690                                          ;*CRAM TEST OF JUMP(I) ON BR7 SET MICRO-PROCESSOR INSTRUCTION.
 4691                                          ;*SET THE BR7 BIT, PERFORM THE JUMP INSTRUCTION.
 4692                                          ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
 4693                                          ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
 4694                                          ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
 4695                                          ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
 4696                                          ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
 4697                                          ;*THEN PORT4 WILL CONTAIN A 37
 4698  017116                                  BADHEAD
 4699                                          ;************** TEST 18 **************
 4700
 4701  017116                          BGNTST
 4702  017116                          T18::
 4703  017116                                  MACEX2                  ;DON'T DO IF M8200.
 4704                                          ;DO NOT DO TEST IF M8200
 4705  017126  104432                          TRAP    C$EXIT
 4706  017130  000230                          .WORD   L10064-.
 4707  017132                                  MYINT
 4708  017132  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
 4709                                                                  ;R1 CONTAINS BASE M8200,4,7 ADDRESS
 4710  017136                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 4711  017142  004737  003474                  JSR     PC,MEMSET       ;SET MEM AND RAM
 4712  017146                          1$:     BGNSEG
 4713  017146  104404                          TRAP    C$BSEG
 4714  017150  004737  003250                  JSR     PC,SETBR7       ;SET THE BR7 BIT'
 4715  017154                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 4716  017154  004537  003100                  JSR     R5,..SROMCLK
 4717  017160  100400                          100400                  ;START AT ROM PC=0
 4718  017162                                  SROMCLK                 ;NEXT WORD IS INSTRUCION,
 4719  017162  004537  003100                  JSR     R5,..SROMCLK
 4720  017166  117777                          114377!<400*7>          ;JUMP TO ROM PC OF 1777
 4721  017170  004737  003330                  JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
 4722  017174  000377                          377                     ;EXPECTED DATA
 4723  017176  120504                          CMPB    R5,R4           ;IS ROM PC CORRECT?
 4724  017200  001406                          BEQ     2$              ;BR IF YES
 4725  017202                                  ERROR   5               ;ERROR, CRAM PC IS WRONG
 4726  017206  104455                          TRAP    C$ERDF
 4727  017210  000005                          .WORD   5
 4728  017212  005055                          .WORD   EM0
 4729  017214  006556                          .WORD   ERR5
 4730  017216                          2$:     ESCAPE SEG
 4731  017216  104410                          TRAP    C$ESCAPE
 4732  017220  000002                          .WORD   10000$-.
 4733  017222                                  ENDSEG
 4734  017222                          10000$:
 4735  017222  104405                          TRAP    C$ESEG
 4736  017224                                  BGNSEG
 4737  017224  104404                          TRAP    C$BSEG
 4738  017226  004737  003250                  JSR     PC,SETBR7               ;SET THE BR7 BIT'
 4739  017232                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 4740  017232  004537  003100                  JSR     R5,..SROMCLK
 4741  017236  100403                          100403                  ;START AT ROM PC=3
```

```
4742  017240
4743  017240  004537  003100          SROMCLK
4744  017244  103400                  JSR     R5,.SROMCLK             ;NEXT WORD IS INSTRUCTION,
4745  017246  004737  003330          100000!<400*7>                  ;JUMP TO ROM PC OF 0
4746  017252  000000                  JSR     PC,RAMDAT               ;R4=CRAM PC (LSB 8 BITS)
4747  017254  120504                  0                               ;EXPECTED DATA
4748  017256  001406                  CMPB    R5,R4                   ;IS ROM PC CORRECT?
4749  017260                          BEQ     4$                      ;BR IF YES
4750  017264  104455                  ERROR   5                       ;ERROR, CRAM PC IS WRONG
4751  017266  000005                  TRAP    C$ERDF
4752  017270  005055                  .WORD   5
4753  017272  006556                  .WORD   EM0
4754  017274                          .WORD   ERR5
4755  017274  104410          4$:     ESCAPE SEG
4756  017276  000002                  TRAP    C$ESCAPE
4757  017300                          .WORD   10001$-.
4758  017300                          ENDSEG
4759  017300  104405          10001$:
4760  017302                          TRAP    C$ESEG
4761  017302  104404                  BGNSEG
4762  017304  004737  003250          TRAP    C$BSEG
4763  017310                          JSR     PC,SETBR7               ;SET THE BR7 BIT'
4764  017310  004537  003100          SROMCLK                         ;NEXT WORD IS INSTRUCTION,
4765  017314  100406                  JSR     R5,.SROMCLK
4766  017316                          100406                          ;START AT ROM PC=6
4767  017316  004537  003100          SROMCLK                         ;NEXT WORD IS INSTRUCTION,
4768  017322  107525                  JSR     R5,.SROMCLK
4769  017324  004737  003330          104125!<400*7>                  ;JUMP TO ROM PC OF 525
4770  017330  000125                  JSR     PC,RAMDAT               ;R4=CRAM PC (LSB 8 BITS)
4771  017332  120504                  125                             ;EXPECTED DATA
4772  017334  001406                  CMPB    R5,R4                   ;IS ROM PC CORRECT?
4773  017336                          BEQ     6$                      ;BR IF YES
4774  017342  104455                  ERROR   5                       ;ERROR, CRAM PC IS WRONG
4775  017344  000005                  TRAP    C$ERDF
4776  017346  005055                  .WORD   5
4777  017350  006556                  .WORD   EM0
4778  017352                          .WORD   ERR5
4779  017352  104410          6$:     ESCAPE SEG
4780  017354  000002                  TRAP    C$ESCAPE
4781  017356                          .WORD   10002$-.
4782  017356                          ENDSEG
4783  017356  104405          10002$:
4784  017360                          TRAP    C$ESEG
4785  017360                  ENDTST
4786  017360  104401          L10064:
4787                                  TRAP    C$ETST
4788  017362
4789                                  BADHEAD
4790                                  ;*************** TEST 19 ***************
4791                                  ;*CRAM TEST OF JUMP(I) ON C BIT CLEAR MICRO-PROCESSOR INSTRUCTION.
4792                                  ;*SET THE C BIT, PERFORM THE JUMP INSTRUCTION.
4793                                  ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
4794                                  ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4795                                  ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4796                                  ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4797                                  ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
                                      ;*THEN PORT4 WILL CONTAIN A 37
```

```
4798  017362                                BADHEAD
4799                                        ;************** TEST 19 **************
4800
4801  017362                          BGNTST
4802  017362                          T19::
4803  017362                                MACEX2                  ;DON'T DO IF M8200.
4804                                        ;DO NOT DO TEST IF M8200
4805  017372  104432                       TRAP    C$EXIT
4806  017374  000244                       .WORD   L10065-.
4807  017376                                MYINT
4808  017376  013701  002516               MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
4809  017402                                MSTCLR                  ;MASTER CLEAR M8200,4,7
4810  017406  004737  003474               JSR     PC,MEMSET       ;SET MEM AND RAM
4811  017412                          1$:   BGNSEG
4812  017412  104404                       TRAP    C$BSEG
4813  017414  004737  003260               JSR     PC,SETC
4814  017420  004737  003166               JSR     PC,CLRALL
4815  017424                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4816  017424  004537  003100               JSR     R5,.SROMCLK
4817  017430  100400                       100400                  ;START AT ROM PC=0
4818  017432                                SROMCLK                 ;NEXT WORD IS INSTRUCION,
4819  017432  004537  003100               JSR     R5,.SROMCLK
4820  017436  115377                       114377!<400*2>          ;JUMP TO ROM PC OF 1777
4821  017440  004737  003330               JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4822  017444  000001                       1                       ;EXPECTED DATA
4823  017446  120504                       CMPB    R5,R4           ;IS ROM PC CORRECT?
4824  017450  001406                       BEQ     2$              ;BR IF YES
4825  017452                                ERROR   5               ;ERROR, CRAM PC IS WRONG
4826  017456  104455                       TRAP    C$ERDF
4827  017460  000005                       .WORD   5
4828  017462  005055                       .WORD   EM0
4829  017464  006556                       .WORD   ERR5
4830  017466                          2$:   ESCAPE SEG
4831  017466  104410                       TRAP    C$ESCAPE
4832  017470  000002                       .WORD   10000$-.
4833  017472                                ENDSEG
4834  017472                          10000$:
4835  017472  104405                       TRAP    C$ESEG
4836  017474                                BGNSEG
4837  017474  104404                       TRAP    C$B[illegible]
4838  017476                                SKIP06  6$
4839                                        ;GOTO 6$ IF M8206
4840  017506  004737  003166               JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
4841  017512                                SROMCLK                 ;NEXT WORD OF INSTRUCTION
4842  017512  004537  003100               JSR     R5,.SROMCLK
4843  017516  100403                       100403                  ;START AT ROM PC=3
4844  017520                                SROMCLK                 ;NEXT WORD OF INSTRUCTION
4845  017520  004537  003100               JSR     R5,.SROMCLK
4846  017524  101000                       100000!<400*2>          ;JUMP TO ROM PC OF 0
4847  017526  004737  003330               JSR     PC,RAMDAT       ;R4=CRAM PC(LSB 8 BITS)
4848  017532  000004                       4                       ;EXPECTED DATA
4849  017534  120504                       CMPB    R5,R4           ;IS ROM PC CORRECT?
4850  017536  001406                       BEQ     4$              ;BR IF YES
4851  017540                                ERROR   5               ;ERROR, CRAM PC IS WRONG
4852  017544  104455                       TRAP    C$ERDF
4853  017546  000005                       .WORD   5
```

```
4854  017550  005055                        .WORD   EM0
4855  017552  006556                        .WORD   ERR5
4856  017554                        4$:     ESCAPE SEG
4857  017554  104410                        TRAP    C$ESCAPE
4858  017556  000002                        .WORD   10001$-.
4859  017560                                ENDSEG
4860  017560                        10001$:
4861  017560  104405                        TRAP    C$ESEG
4862  017562                                BGNSEG
4863  017562  104404                        TRAP    C$BSEG
4864  017564  004737  003166                JSR     PC,CLRALL               ;CLEAR ALL CONDITIONS
4865  017570                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4866  017570  004537  003100                JSR     R5,.SROMCLK
4867  017574  100406                        100406                  ;START AT ROM PC=6
4868  017576                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4869  017576  004537  003100                JSR     R5,.SROMCLK
4870  017602  105125                        104125!<400*2>          ;JUMP TO ROM PC OF 525
4871  017604  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4872  017610  000007                        7                       ;EXPECTED DATA
4873  017612  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
4874  017614  001406                        BEQ     6$              ;BR IF YES
4875  017616                                ERROR   5               ;ERROR, CRAM PC IS WRONG
4876  017622  104455                        TRAP    C$ERDF
4877  017624  000005                        .WORD   5
4878  017626  005055                        .WORD   EM0
4879  017630  006556                        .WORD   ERR5
4880  017632                        6$:     ESCAPE SEG
4881  017632  104410                        TRAP    C$ESCAPE
4882  017634  000002                        .WORD   10002$-.
4883  017636                                ENDSEG
4884  017636                        10002$:
4885  017636  104405                        TRAP    C$ESEG
4886  017640                        ENDTST
4887  017640                        L10065:
4888  017640  104401                        TRAP    C$ETST
4889
4890  017642                                BADHEAD
4891                                        ;************** TEST 20 **************
4892                                        ;*CRAM TEST OF JUMP(I) ON Z BIT CLEAR MICRO-PROCESSOR INSTRUCTION.
4893                                        ;*CLEAR THE Z BIT, PERFORM THE JUMP INSTRUCTION.
4894                                        ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
4895                                        ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4896                                        ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4897                                        ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4898                                        ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
4899                                        ;*THEN PORT4 WILL CONTAIN A 37
4900  017642                                BADHEAD
4901                                        ;************** TEST 20 **************
4902
4903  017642                        BGNTST
4904  017642                        T20::
4905  017642                                MACEX2                  ;DON'T DO IF M8200.
4906                                        ;DO NOT DO TEST IF M9200
4907  017652  104432                        TRAP    C$EXIT
4908  017654  000244                        .WORD   L10066-.
4909  017656                                MYINT
```

```
4910  017656  013701  002516          MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
4911  017662                          MSTCLR                  ;MASTER CLEAR M8200,4,7
4912  017666  004737  003474          JSR     PC,MEMSET       ;SET MEM AND RAM
4913  017672                  1$:     BGNSEG
4914  017672  104404                  TRAP    C$BSEG
4915  017674  004737  003312          JSR     PC,SETZ
4916  017700  004737  003166          JSR     PC,CLRALL       ; CLEAR CONDITION CODES ;*** B0
4917  017704                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4918  017704  004537  003100          JSR     R5,..SROMCLK
4919  017710  100400                  100400                  ;START AT ROM PC=0
4920  017712                          SROMCLK                 ;NEXT WORD IS INSTRUCION,
4921  017712  004537  003100          JSR     R5,..SROMCLK
4922  017716  115777                  114377!<400*3>          ;JUMP TO ROM PC OF 1777
4923  017720  004737  003330          JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4924  017724  000001                  1                       ;EXPECTED DATA
4925  017726  120504                  CMPB    R5,R4           ;IS ROM PC CORRECT?
4926  017730  001406                  BEQ     2$              ;BR IF YES
4927  017732                          ERROR   5               ;ERROR, CRAM PC IS WRONG
4928  017736  104455                  TRAP    C$ERDF
4929  017740  000005                  .WORD   5
4930  017742  005055                  .WORD   EM0
4931  017744  006556                  .WORD   ERR5
4932  017746                  2$:     ESCAPE SEG
4933  017746  104410                  TRAP    C$ESCAPE
4934  017750  000002                  .WORD   10000$-.
4935  017752                          ENDSEG
4936  017752                  10000$:
4937  017752  104405                  TRAP    C$ESEG
4938  017754                          BGNSEG
4939  017754  104404                  TRAP    C$BSEG
4940  017756                          SKIP06  6$
4941                                  ;GOTO 6$ IF M8206
4942  017766  004737  003166          JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
4943  017772                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4944  017772  004537  003100          JSR     R5,..SROMCLK
4945  017776  100403                  100403                  ;START AT ROM PC=3
4946  020000                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4947  020000  004537  003100          JSR     R5,..SROMCLK
4948  020004  101400                  100000!<400*3>          ;JUMP TO ROM PC OF 0
4949  020006  004737  003330          JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4950  020012  000004                  4                       ;EXPECTED DATA
4951  020014  120504                  CMPB    R5,R4           ;IS ROM PC CORRECT?
4952  020016  001406                  BEQ     4$              ;BR IF YES
4953  020020                          ERROR   5               ;ERROR, CRAM PC IS WRONG
4954  020024  104455                  TRAP    C$ERDF
4955  020026  000005                  .WORD   5
4956  020030  005055                  .WORD   EM0
4957  020032  006556                  .WORD   ERR5
4958  020034                  4$:     ESCAPE SEG
4959  020034  104410                  TRAP    C$ESCAPE
4960  020036  000002                  .WORD   10001$-.
4961  020040                          ENDSEG
4962  020040                  10001$:
4963  020040  104405                  TRAP    C$ESEG
4964  020042                          BGNSEG
4965  020042  104404                  TRAP    C$BSEG
```

```
 4966  020044  004737  003166                 JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
 4967  020050                                 SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 4968  020050  004537  003100                 JSR     R5,.SROMCLK
 4969  020054  100406                         100406                  ;START AT ROM PC=6
 4970  020056                                 SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 4971  020056  004537  003100                 JSR     R5,.SROMCLK
 4972  020062  105525                         104125!<400*3>          ;JUMP TO ROM PC OF 525
 4973  020064  004737  003330                 JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
 4974  020070  000007                         7                       ;EXPECTED DATA
 4975  020072  120504                         CMPB    R5,R4           ;IS ROM PC CORRECT?
 4976  020074  001406                         BEQ     6$              ;BR IF YES
 4977  020076                                 ERROR   5               ;ERROR, CRAM PC IS WRONG
 4978  020102  104455                         TRAP    C$ERDF
 4979  020104  000005                         .WORD   5
 4980  020106  005055                         .WORD   EMO
 4981  020110  006556                         .WORD   ERR5
 4982  020112                         6$:     ESCAPE SEG
 4983  020112  104410                         TRAP    C$ESCAPE
 4984  020114  000002                         .WORD   10002$-.
 4985  020116                                 ENDSEG
 4986  020116                         10002$:
 4987  020116  104405                         TRAP    C$ESEG
 4988  020120                         ENDTST
 4989  020120                         L10066:
 4990  020120  104401                         TRAP    C$ETST
 4991
 4992  020122                                 BADHEAD
 4993                                         ;************** TEST 21 **************
 4994                                         ;*CRAM TEST OF JUMP(I) ON BRO CLEAR MICRO-PROCESSOR INSTRUCTION.
 4995                                         ;*CLEAR THE BRO BIT, PERFORM THE JUMP INSTRUCTION.
 4996                                         ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
 4997                                         ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
 4998                                         ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
 4999                                         ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
 5000                                         ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
 5001                                         ;*THEN PORT4 WILL CONTAIN A 37
 5002  020122                                 BADHEAD
 5003                                         ;************** TEST 21 **************
 5004
 5005  020122                         BGNTST
 5006  020122                         T21::
 5007  020122                                 MACEX2                  ;DON'T DO IF M8200.
 5008                                         ;DO NOT DO TEST IF M8200
 5009  020132  104432                         TRAP    C$EXIT
 5010  020134  000240                         .WORD   L10067-.
 5011  020136                                 MYINT
 5012  020136  013701  002516                 MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
 5013  020142                                 MSTCLR                  ;MASTER CLEAR M8200,4,7
 5014  020146  004737  003474                 JSR     PC,MEMSET       ;SET MEM AND RAM
 5015  020152                         1$:     BGNSEG
 5016  020152  104404                         TRAP    C$BSEG
 5017  020154  004737  003166                 JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
 5018  020160                                 SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 5019  020160  004537  003100                 JSR     R5,.SROMCLK
 5020  020164  100400                         100400                  ;START AT ROM PC=0
 5021  020166                                 SROMCLK                 ;NEXT WORD IS INSTRUCION,
```

```
5022  020166  004537  003100                JSR     R5,.SROMCLK
5023  020172  116377                        114377!<400*4>          ;JUMP TO ROM PC OF 1777
5024  020174  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5025  020200  000001                        1                       ;EXPECTED DATA
5026  020202  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
5027  020204  001406                        BEQ     2$              ;BR IF YES
5028  020206                                ERROR   5               ;ERROR, CRAM PC IS WRONG
5029  020212  104455                        TRAP    C$ERDF
5030  020214  000005                        .WORD   5
5031  020216  005055                        .WORD   EM0
5032  020220  006556                        .WORD   ERR5
5033  020222                        2$:     ESCAPE SEG
5034  020222  104410                        TRAP    C$ESCAPE
5035  020224  000002                        .WORD   10000$-.
5036  020226                                ENDSEG
5037  020226                        10000$:
5038  020226  104405                        TRAP    C$ESEG
5039  020230                                BGNSEG
5040  020230  104404                        TRAP    C$BSEG
5041  020232                                SKIP06  6$
5042                                        ;GOTO 6$ IF M8206
5043  020242  004737  003166                JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5044  020246                                SROMCLK                 ;NEXT WORD IS INSTRUCTION.
5045  020246  004537  003100                JSR     R5,.SROMCLK
5046  020252  100403                        100403                  ;START AT ROM PC-3
5047  020254                                SROMCLK                 ;NEXT WORD IS INSTRUCTION.
5048  020254  004537  003100                JSR     R5,.SROMCLK
5049  020260  102000                        100000!<400*4>          ;JUMP TO ROM PC OF 0
5050  020262  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5051  020266  000004                        4                       ;EXPECTED DATA
5052  020270  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
5053  020272  001406                        BEQ     4$              ;BR IF YES
5054  020274                                ERROR   5               ;ERROR, CRAM PC IS WRONG
5055  020300  104455                        TRAP    C$ERDF
5056  020302  000005                        .WORD   5
5057  020304  005055                        .WORD   EM0
5058  020306  006556                        .WORD   ERR5
5059  020310                        4$:     ESCAPE SEG
5060  020310  104410                        TRAP    C$ESCAPE
5061  020312  000002                        .WORD   10001$-.
5062  020314                                ENDSEG
5063  020314                        10001$:
5064  020314  104405                        TRAP    C$ESEG
5065  020316                                BGNSEG
5066  020316  104404                        TRAP    C$BSEG
5067  020320  004737  003166                JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5068  020324                                SROMCLK                 ;NEXT WORD IS INSTRUCTION.
5069  020324  004537  003100                JSR     R5,.SROMCLK
5070  020330  100406                        100406                  ;START AT ROM PC=6
5071  020332                                SROMCLK                 ;NEXT WORD IS INSTRUCTION.
5072  020332  004537  003100                JSR     R5,.SROMCLK
5073  020336  106125                        104125!<400*4>          ;JUMP TO ROM PC OF 525
5074  020340  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5075  020344  000007                        7                       ;EXPECTED DATA
5076  020346  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
5077  020350  001406                        BEQ     6$              ;BR IF YES
```

```
5078  020352
5079  020356  104455                    ERROR   5               ;ERROR, CRAM PC IS WRONG
5080  020360  000005                    TRAP    C$ERDF
5081  020362  005055                    .WORD   5
5082  020364  006556                    .WORD   EM0
5083  020366                            .WORD   ERR5
5084  020366  104410              6$:   ESCAPE SEG
5085  020370  000002                    TRAP    C$ESCAPE
5086  020372                            .WORD   10002$-.
5087  020372                            ENDSEG
5088  020372  104405              10002$:
5089  020374                            TRAP    C$ESEG
5090  020374                      ENDTST
5091  020374  104401              L10067:
5092                                    TRAP    C$ETST
5093  020376
5094                                    BADHEAD
5095                                    ;************** TEST 22 **************
5096                                    ;*CRAM TEST OF JUMP(I) ON BR1 CLEAR MICRO-PROCESSOR INSTRUCTION.
5097                                    ;*CLEAR THE BR1 BIT, PERFORM THE JUMP INSTRUCTION.
5098                                    ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
5099                                    ;*IN THE LOCATION IT IS AT. THIS I STRUCTION LOADS THE
5100                                    ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
5101                                    ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
5102                                    ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
5103  020376                            ;*THEN PORT4 WILL CONTAIN A 37
5104                                    BADHEAD
5105                                    ;************** TEST 22 **************
5106  020376
5107  020376                      BGNTST
5108  020376                      T22::
5109                                    MACEX2                  ;DON'T DO IF M8200.
5110  020406  104432                    ;DO NOT DO TEST IF M8200
5111  020410  000240                    TRAP    C$EXIT
5112  020412                            .WORD   L10070-.
5113  020412  013701  002516            MYINT
5114  020416                            MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
5115  020422  004737  003474            MSTCLR                  ;MASTER CLEAR M8200,4,7
5116  020426                            JSR     PC,MEMSET       ;SET MEM AND RAM
5117  020426  104404              1$:   BGNSEG
5118  020430  004737  003166            TRAP    C$BSEG
5119  020434                            JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5120  020434  004537  003100            SROMCLK                 ;NEXT WORD IS INSTRUCTION.
5121  020440  100400                    JSR     R5,.SROMCLK
5122  020442                            100400                  ;START AT ROM PC=0
5123  020442  004537  003100            SROMCLK                 ;NEXT WORD IS INSTRUCION.
5124  020446  116777                    JSR     R5,.SROMCLK
5125  020450  004737  003330            114377!<400*5>          ;JUMP TO ROM PC OF 1777
5126  020454  000001                    JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5127  020456  120504                    1                       ;EXPECTED DATA
5128  020460  001406                    CMPB    R5,R4           ;IS ROM PC CORRECT?
5129  020462                            BEQ     2$              ;BR IF YES
5130  020466  104455                    ERROR   5               ;ERROR, CRAM PC IS WRONG
5131  020470  000005                    TRAP    C$ERDF
5132  020472  005055                    .WORD   5
5133  020474  006556                    .WORD   EM0
                                        .WORD   ERR5
```

```
5134  020476                        2$:     ESCAPE SEG
5135  020476  104410                        TRAP    C$ESCAPE
5136  020500  000002                        .WORD   10000$-.
5137  020502                                ENDSEG
5138  020502                        10000$:
5139  020502  104405                        TRAP    C$ESEG
5140  020504                                BGNSEG
5141  020504  104404                        TRAP    C$BSEG
5142  020506                                SKIP06  6$
5143                                        ;GOTO 6$ IF M8206
5144  020516  004737  003166                JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5145  020522                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5146  020522  004537  003100                JSR     R5,.SROMCLK
5147  020526  100403                        100403                  ;START AT ROM PC=3
5148  020530                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5149  020530  004537  003100                JSR     R5,.SROMCLK
5150  020534  102400                        100000!<400*5>          ;JUMP TO ROM PC OF 0
5151  020536  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5152  020542  000004                        4                       ;EXPECTED DATA
5153  020544  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
5154  020546  001406                        BEQ     4$              ;BR IF YES
5155  020550                                ERROR   5               ;ERROR, CRAM PC IS WRONG
5156  020554  104455                        TRAP    C$ERDF
5157  020556  000005                        .WORD   5
5158  020560  005055                        .WORD   EM0
5159  020562  006556                        .WORD   ERR5
5160  020564                        4$:     ESCAPE SEG
5161  020564  104410                        TRAP    C$ESCAPE
5162  020566  000002                        .WORD   10001$-.
5163  020570                                ENDSEG
5164  020570                        10001$:
5165  020570  104405                        TRAP    C$ESEG
5166  020572                                BGNSEG
5167  020572  104404                        TRAP    C$BSEG
5168  020574  004737  003166                JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5169  020600                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5170  020600  004537  003100                JSR     R5,.SROMCLK
5171  020604  100406                        100406                  ;START AT ROM PC=6
5172  020606                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5173  020606  004537  003100                JSR     R5,.SROMCLK
5174  020612  106525                        104125!<400*5>          ;JUMP TO ROM PC OF 525
5175  020614  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5176  020620  000007                        7                       ;EXPECTED DATA
5177  020622  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
5178  020624  001406                        BEQ     6$              ;BR IF YES
5179  020626                                ERROR   5               ;ERROR, CRAM PC IS WRONG
5180  020632  104455                        TRAP    C$ERDF
5181  020634  000005                        .WORD   5
5182  020636  005055                        .WORD   EM0
5183  020640  006556                        .WORD   ERR5
5184  020642                        6$:     ESCAPE SEG
5185  020642  104410                        TRAP    C$ESCAPE
5186  020644  000002                        .WORD   10002$-.
5187  020646                                ENDSEG
5188  020646                        10002$:
5189  020646  104405                        TRAP    C$ESEG
```

```
5190  020650                          ENDTST
5191  020650                          L10070:
5192  020650  104401                          TRAP    C$ETST
5193
5194  020652                          BADHEAD
5195                                  ;************** TEST 23 **************
5196                                  ;*CRAM TEST OF JUMP(I) ON BR4 CLEAR MICRO-PROCESSOR INSTRUCTION.
5197                                  ;*CLEAR THE BR4 BIT, PERFORM THE JUMP INSTRUCTION.
5198                                  ;*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
5199                                  ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
5200                                  ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
5201                                  ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
5202  020652  020652                  .
5203                                  ;*THE CRAM PC IS CORRECT, IF THE CRAM PC IS NOT RIGHT.
5204                                  ;*THEN PORT4 CONTAINS A 37
5205  020654                          BADHEAD
5206                                  ;************** TEST 23 **************
5207
```

```
5208  020654                          BGNTST
5209  020654                          T23::
5210  020654                                  MACEX2                  ;DON'T DO IF M8200.
5211                                          ;DO NOT DO TEST IF M8200
5212  020664  104432                          TRAP    C$EXIT
5213  020666  000240                          .WORD   L10071-.
5214  020670                                  MYINT
5215  020670  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
5216  020674                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
5217  020700  004737  003474                  JSR     PC,MEMSET       ;SET MEM AND RAM
5218  020704                          1$:     BGNSEG
5219  020704  104404                          TRAP    C$BSEG
5220  020706  004737  003166                  JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5221  020712                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION.
5222  020712  004537  003100                  JSR     R5,..SROMCLK
5223  020716  100400                          100400                  ;START AT ROM PC=0
5224  020720                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION.
5225  020720  004537  003100                  JSR     R5,..SROMCLK
5226  020724  117377                          114377!<400*6>          ;JUMP TO ROM PC OF 1777
5227  020726  004737  003330                  JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
```

```
5228  020732  000001                        1                       ;EXPECTED DATA
5229  020734  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
5230  020736  001406                        BEQ     2$              ;BR IF YES
5231  020740                                ERROR   5               ;ERROR, CRAM PC IS WRONG
5232  020744  104455                        TRAP    C$ERDF
5233  020746  000005                        .WORD   5
5234  020750  005055                        .WORD   EM0
5235  020752  006556                        .WORD   ERR5
5236  020754                          2$:   ESCAPE SEG
5237  020754  104410                        TRAP    C$ESCAPE
5238  020756  000002                        .WORD   10000$-.
5239  020760                                ENDSEG
5240  020760                          10000$:
5241  020760  104405                        TRAP    C$ESEG
5242  020762                                BGNSEG
5243  020762  104404                        TRAP    C$BSEG
5244  020764                                SKIP06  6$
5245                                        ;GOTO 6$ IF M8206
5246  020774  004737  003166                JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5247  021000                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5248  021000  004537  003100                JSR     R5,..SROMCLK
5249  021004  100403                        100403                  ;START AT ROM PC=3
5250  021006                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5251  021006  004537  003100                JSR     R5,..SROMCLK
5252  021012  103000                        100000!<400*6>  ;JUMP TO ROM PC OF 0
5253  021014  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5254  021020  000004                        4                       ;EXPECTED DATA
```

```
5255  021022  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
5256  021024  001406                        BEQ     4$              ;BSR IF YES
```

```
 5257  021026                                ERROR   5               ;ERROR, CRAM PC IS WRONG
 5258  021032  104455                        TRAP    C$ERDF
 5259  021034  000005                        .WORD   5
 5260  021036  005055                        .WORD   EM0
 5261  021040  006556                        .WORD   ERR5
```

```
5262  021042
5263  021042  104410                4$:     ESCAPE SEG
5264  021044  000002                        TRAP    C$ESCAPE
5265  021046                                .WORD   10001$-.
5266  021046                                ENDSEG
5267  021046  104405                10001$:
5268  021050                                TRAP    C$ESEG
5269  021050  104404                        BGNSEG
5270  021052  004737  003166                TRAP    C$BSEG
5271  021056                                JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5272  021056  004537  003100                SROMCLK                 ;NEXT WORD IS INSTRUCTION.
5273  021062  100406                        JSR     R5,..SROMCLK
5274  021064                                100406                  ;START AT ROM PC=6
5275  021064  004537  003100                SROMCLK                 ;NEXT WORD IS INSTRUCTION.
5276  021070  107125                        JSR     R5,..SROMCLK
5277  021072  004737  003330                104125!<400*6>  ;JUMP TO ROM PC OF 525
5278  021076  000007                        JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5279  021100  120504                        7                       ;EXPECTED DATA
5280  021102  001406                        CMPB    R5,R4           ;IS ROM PC CORRECT?
5281  021104                                BEQ     6$              ;BR IF YES
5282  021110  104455                        ERROR   5               ;EPROR, CRAM PC IS WRONG
5283  021112  000005                        TRAP    C$ERDF
5284  021114  005055                        .WORD   5
5285  021116  006556                        .WORD   EM0
5286  021120                                .WORD   ERR5
5287  021120  104410                6$:     ESCAPE SEG
5288  021122  000002                        TRAP    C$ESCAPE
5289  021124                                .WORD   10002$-.
5290  021124                                ENDSEG
5291  021124  104405                10002$:
5292  021126                                TRAP    C$ESEG
5293  021126                        ENDTST
5294  021126  104401                L10071:
5295                                        TRAP    C$ETST
5296  021130                                BADHEAD
5297                                        :************** TEST 24 **************
5298                                        :*CRAM TEST OF JUMP(I) ON BR7 CLEAR MICRO-PROCESSOR INSTRUCTION.
5299                                        :*CLEAR THE BR7 BIT, PERFORM THE JUMP INSTRUCTION.
5300                                        :*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
5301                                        :*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
5302                                        :*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
5303                                        :*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
5304                                        :*THE CRAM PC IS CORRECT, IF THE CRAM PC IS NOT RIGHT.
5305                                        :*THEN PORT4 CONTAINS A 37
5306  021130                                BADHEAD
5307                                        :************** TEST 24 **************
5308
5309  021130                        BGNTST
5310  021130                        T24::
5311  021130                                MACEX2                  ;DON'T DO IF M8200.
5312                                        ;DO NOT DO TEST IF M8200
5313  021140  104432                        TRAP    C$EXIT
5314  021142  000240                        .WORD   L10072-.
5315  021144                                MYINT
5316  021144  013701  002516                MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
5317  021150                                MSTCLR                  ;MASTER CLEAR M8200,4,7
```

```
5318  021154  004737  003474          JSR     PC,MEMSET       ;SET MEM AND RAM
5319  021160                  1$:     BGNSEG
5320  021160  104404                  TRAP    C$BSEG
5321  021162  004737  003166          JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5322  021166                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5323  021166  004537  003100          JSR     R5,.SROMCLK
5324  021172  100400                  100400          ;START AT ROM PC=0
5325  021174                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5326  021174  004537  003100          JSR     R5,.SROMCLK
5327  021200  117777                  114377!<400*7>          ;JUMP TO ROM PC OF 1777
5328  021202  004737  003330          JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5329  021206  000001                  1                       ;EXPECTED DATA
5330  021210  120504                  CMPB    R5,R4           ;IS ROM PC CORRECT?
5331  021212  001406                  BEQ     2$              ;BR IF YES
5332  021214                          ERROR   5               ;ERROR, CRAM PC IS WRONG
5333  021220  104455                  TRAP    C$ERDF
5334  021222  000005                  .WORD   5
5335  021224  005055                  .WORD   EM0
5336  021226  006556                  .WORD   ERR5
5337  021230                  2$:     ESCAPE SEG
5338  021230  104410                  TRAP    C$ESCAPE
5339  021232  000002                  .WORD   10000$-.
5340  021234                          ENDSEG
5341  021234                  10000$:
5342  021234  104405                  TRAP    C$ESEG
5343  021236                          BGNSEG
5344  021236  104404                  TRAP    C$BSEG
5345  021240                          SKIP06  6$
5346                                  ;GOTO 6$ IF M8206
5347  021250  004737  003166          JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5348  021254                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5349  021254  004537  003100          JSR     R5,.SROMCLK
5350  021260  100403                  100403                  ;START AT ROM PC=3
5351  021262                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5352  021262  004537  003100          JSR     R5,.SROMCLK
5353  021266  103400                  100000!<400*7>  ;JUMP TO ROM PC OF 0
5354  021270  004737  003330          JSR     PC,RAMDAT       ;;R4=CRAM PC (LCB 8 BITS)
5355  021274  000004                  4                       ;EXPECTED DATA
5356  021276  120504                  CMPB    R5,R4           ;IS ROM PC CORRECT?
5357  021300  001406                  BEQ     4$              ;BR IF YES
5358  021302                          ERROR   5               ;ERROR, CRAM PC IS WRONG
5359  021306  104455                  TRAP    C$ERDF
5360  021310  000005                  .WORD   5
5361  021312  005055                  .WORD   EM0
5362  021314  006556                  .WORD   ERR5
5363  021316                  4$:     ESCAPE SEG
5364  021316  104410                  TRAP    C$ESCAPE
5365  021320  000002                  .WORD   10001$-.
5366  021322                          ENDSEG
5367  021322                  10001$:
5368  021322  104405                  TRAP    C$ESEG
5369  021324                          BGNSEG
5370  021324  104404                  TRAP    C$BSEG
5371  021326  004737  003166          JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5372  021332                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5373  021332  004537  003100          JSR     R5,.SROMCLK
```

```
5374  021336  100406                        100406                  ;START AT ROM PC=6
5375  021340                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5376  021340  004537  003100                JSR     R5,.SROMCLK
5377  021344  107525                        104125!<400*7>          ;JUMP TO ROM PC OF 525
5378  021346  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5379  021352  000007                        7                       ;EXPECTED DATA
5380  021354  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
5381  021356  001406                        BEQ     6$              ;BR IF YES
5382  021360                                ERROR   5               ;ERROR, CRAM PC IS WRONG
5383  021364  104455                        TRAP    C$ERDF
5384  021366  000005                        .WORD   5
5385  021370  005055                        .WORD   EM0
5386  021372  006556                        .WORD   ERR5
5387  021374                        6$:     ESCAPE SEG
5388  021374  104410                        TRAP    C$ESCAPE
5389  021376  000002                        .WORD   10002$-.
5390  021400                                ENDSEG
5391  021400                        10002$:
5392  021400  104405                        TRAP    C$ESEG
5393  021402                        ENDTST
5394  021402                        L10072:
5395  021402  104401                        TRAP    C$ETST
5396
5397  021404                                BADHEAD
5398                                        ;************** TEST 25 **************
5399                                        ;*
5400                                        ;*MAIN MEMORY PAGE DUAL ADDRESS TEST.
5401                                        ;*IN THIS TEST WE WILL VERIFY THAT PAGES DO
5402                                        ;*NOT DUAL ADDRESS.  THIS TEST IS DIFFERENT FROM THE
5403                                        ;*PREVIOUS DUAL ADDRESS TESTS IN THAT THE OTHER
5404                                        ;*TEST REALLY DIDN'T CHECK PAGE DUAL ADDRESSING
5405  021404                                BADHEAD
5406                                        ;************** TEST 25 **************
5407
5408  021404                        BGNTST
5409  021404                        T25::
5410  021404                                K4ONLY                  ;FOR 4K CPUS ONLY.
5411                                        ;DO NOT DO TEST IF M8200, OR M8204
5412  021414  104432                        TRAP    C$EXIT
5413  021416  000156                        .WORD   L10073-.
5414  021420                                MYINT
5415  021420  013701  002516                MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
5416  021424                                MSTCLR
5417  021430  005002                        CLR R2                  ;R2 WILL BE PAGE #
5418  021432  042737  000037  021456  1$:   BIC #37,2$             ;CLEAR UNUSED BITS
5419  021440  050237  021456                BIS R2,2$               ;ADD CURRENT PAGE MARKER.
5420  021444                                ROMCLK                  ;SET ADDR D
5421  021444  004537  003044                JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
5422  021450  010000                        10000
5423  021452                                ROMCLK                  ;OF PAGE X
5424  021452  004537  003044                JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
5425  021456  004000                2$:     4000                    ;THIS LOCATION MODIFIED BY LOST
5426                                                                ;FEW INSTRUCTIONS
5427  021460  010261  000004                MOV R2,4(R1)            ;PUT PAGE # INTO PART 4
5428  021464                                ROMCLK                  ;CLOCK PART 4 INTO MEMORY
5429  021464  004537  003044                JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
```

```
5430  021470  122500                          122500                  ;WHOSE PAGE # IS IN R2
5431  021472  005202                          INC R2                  ;UPDATE PAGE #
5432  021474  032702  000020                  BIT #20,R2              ;DONE ALL PAGES?
5433  021500  001754                          BEQ 1$                  ;NO-DO NEXT ONE
5434
5435                                                                  ;OK, ALL LOADS OF ALL PAGES
5436                                                                  ;CONTAIN THIER PAGE NUMBER, NOW
5437                                                                  ;WE'LL GO BACK THROUGH
5438                                                                  ;THEM CHECKING THEM OUT.
5439
5440  021502  005002                          CLR R2                  ;R2 STILL HAS PAGE NUMBER
5441
5442  021504  042737  000037  021522  3$:     BIC #37,4$
5443  021512  050237  021522                  BIS R2,4$
5444  021516                                  ROMCLK                  ;LOAD PAGE NUMBER
5445  021516  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
5446  021522  004000                  4$:     4000
5447  021524                                  ROMCLK                  ;MOVE MEM TO PART 4
5448  021524  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
5449  021530  041224                          041224
5450  021532  116104  000004                  MOVB 4(R1),R4           ;''FOUND''
5451  021536  110205                          MOVB R2,R5              ;''EXPECTED''
5452  021540  120504                          CMPB R5,R4              ;ADDRESS PROBLEM?
5453  021542  001406                          BEQ 5$
5454
5455  021544                                  ERROR 13                ;PAGE ADDRESSING ERROR IN MAIN
5456  021550  104455                          TRAP    C$ERDF
5457  021552  000015                          .WORD   13
5458  021554  005055                          .WORD   EM0
5459  021556  007542                          .WORD   ERR13
5460                                                                  ;MEMORY.
5461                                                                  ;MAR BITS 8 THROUGH 12 ARE REPRESENTED
5462                                                                  ;BY R2 (''EXP'ED'')BITS 0-4)
5463
5464  021560                          5$:     ESCAPE TST
5465  021560  104410                          TRAP    C$ESCAPE
5466  021562  000012                          .WORD   L10073-.
5467  021564  005202                          INC R2                  ;UPDATE PAGE ADDRESS
5468  021566  032702  000020                  BIT #20,R2              ;ALL DONE?
5469  021572  001744                          BEQ 3$                  ;NO-CHECK NEXT PAGE.
5470                                                                  ;YES-EXIT.
5471  021574                          ENDTST
5472  021574                          L10073:
5473  021574  104401                          TRAP    C$ETST
5474
5475
5476  021576                                  BADHEAD
5477                                          ;************** TEST 26 **************
5478                                          ;*
5479                                          ;*JUMP FIELD,PAGE TEST
5480                                          ;*
5481                                          ;*IN THIS TEST WILL MAKE SURE A JUMP FIELD INSTRUCTION
5482                                          ;*WORKS.  TO DO THIS, WE'LL PUT THE DESIRED PAGE,FIELD
5483                                          ;*INORMATION IN IBUS*<13> THEN ISSUE A JUMP FIELD
5484                                          ;*THEN WE'LL READ PC REG. AND VERIFY.
5485                                          ;** REV. E - CHANGE TO TEST BITS 4 & 5 (PC BITS 12 & 13)
```

```
5486  021576                                  RADHEAD
5487                                          ;************** TEST 26 **************
5488
5489  021576                            BGNTST
5490  021576                            T26::
5491  021576                                  K4ONLY                  ;FOR 4K CPUS ONLY
5492                                          ;DO NOT DO TEST IF M8200, OR M8204
5493  021606  104432                         TRAP    C$EXIT
5494  021610  000132                         .WORD   L10074-.
5495  021612                                  MYINT
5496  021612  013701  002516                 MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
5497  021616                                  MSTCLR
5498
5499  021622  005002                         CLR R2                  ;R2 TO CONTAIN FIELD #
5500
5501
5502
5503  021624  042737  000077  021642  1$:   BIC #77,2$               ;CLEAR ANY JUNK (REV E - CHANGE TO 77)
5504  021632  050237  021642                 BIS R2,2$               ;SET FIELD # INTO INSTR.
5505
5506  021636                                  ROMCLK                  ;CLOCK FIELD BITS INTO BREG.
5507  021636  004537  003044                 JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
5508  021642  000400                   2$:   000400                  ;CONTAINS FIELD,PAGE BITS
5509  021644                                  ROMCLK                  ;XFERR BREG INTO IBUS*<13>
5510  021644  004537  003044                 JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
5511  021650  061233                         061233
5512  021652                                  SROMCLK                 ;GET INSTRUCTION CLOCKED.
5513  021652  004537  003100                 JSR     R5,..SROMCLK
5514  021656  100000                         100000                  ;BAS FORM FOR JUM FIELD INSTR.
5515
5516
5517  021660  142761  000002  000001         BICB #BIT1,1(R1)                ;CLEAR ROMI
5518  021666                                  ROMCLK                  ;CLOCK NEXT INSTR.
5519  021666  004537  003044                 JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
5520  021672  121264                         121264                  ;MOVE IBUS*TO PORT 4
5521  021674  116104  000004                 MOVB 4(R1),R4           ;GET IT.
5522  021700  042704  177700                 BIC     #^C<77>,R4      ;(REV E - CHANGE TO 77)
5523  021704  120402                         CMPB R4,R2              ;FIELD OK?
5524  021706  001407                         BEQ 3$                  ;IF OK GO AHEAD
5525  021710  010205                         MOV     R2,R5
5526  021712                                  ERROR 14                ;CHANGE FIELD INSTRUCTION   ;*** B0
5527  021716  104455                         TRAP    C$ERDF
5528  021720  000016                         .WORD   14
5529  021722  005055                         .WORD   EM0
5530  021724  007664                         .WORD   ERR14
5531                                                                  ;FAILED.  FOR FIELD,PAGE INDICATES
5532                                                                  ;BY "EXPECTED" BITS 0,1,2,3 OF
5533                                                                  ;EXPECTED REPRESENT FIELD BITS.
5534  021726                            3$:   ESCAPE TST
5535  021726  104410                         TRAP    C$ESCAPE
5536  021730  000012                         .WORD   L10074-.
5537
5538
5539  021732  005202                         INC R2                  ;UPDATE TO NEXT FIELD
5540  021734  032702  000100                 BIT #100,R2             ;DONE ALL FIELDS? (REV E - CHANGE TO 100)
5541  021740  001731                         BEQ     1$
```

```
5542
5543  021742                                ENDTST
5544  021742                                L10074:
5545  021742  104401                              TRAP    C$ETST
5546
5547  021744                                BADHEAD
5548                                        ;************** TEST 27 **************
5549                                        ;*
5550                                        ;*JUMP TEST, JUMP ALWAYS, JUMP CHANGE FIELD
5551                                        ;*
5552                                        ;*IN THIS TEST, WE WILL CHECK THE ABILITY OF THE
5553                                        ;*MICRO PROCESSOR TO JUMP (BRANCH & ALWAYS INSTRUCTION)
5554                                        ;*TO LOCATIONS, FIELDS FROM OTHER LOCATIONS FIELDS.
5555                                        ;*WE ALREADY KNOW THAT THE BRANCH INSTR WORKS FROM
5556                                        ;*OTHER TEST.                PROCEDURE:
5557                                        ;*      1. START ADDR 0, FIELD 0
5558                                        ;*      2. **CALCULATE NEW ADDRS, FIELD  VIA INC,
5559                                        ;*      3. CAUSE JUMP (BRANCH) TO NEW ADDRESS
5560                                        ;*      4. READ PC FROM IBUS*12 AND IBUS*13
5561                                        ;*      5. REPEAT STEP 2-4 256.TIMES
5562                                        ;*
5563                                        ;*      TO CALCULATE NEW ADDRESS:
5564                                        ;*      1. INC LOW BYTE OF ADDRESS FOR PC ADDRES 0-7
5565                                        ;*      2. INC LOW BYTE OF NADDRESS FOR PC ADDRESS 8-11
5566                                        ;*         BITS REPRESENTED AS BITS 0-3. WHEN  0-3 OVERFLOWS,
5567                                        ;*         RESTARTS AT ZERO.
5568                                        ;*         NET RESULT IS JUMPS FROM:
5569                                        ;*              FIELD,PAGE              LOC
5570                                        ;*                   0                  0
5571                                        ;*                   1                  1
5572                                        ;*                   2                  2
5573                                        ;*                   3                  3
5574                                        ;*                  10                  7
5575                                        ;*                  11                 11
5576                                        ;*                  :10                 :
5577                                        ;*                  17                377
5578                                        ;*
5579  021744                                BADHEAD
5580                                        ;************** TEST 27 **************
5581
5582  021744                                BGNTST
5583  021744                          T27::
5584  021744                                MYINT
5585  021744  013701  002516                MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
5586  021750                                K4ONLY                          ;4K CPUS ONLY.
5587                                        ;DO NOT DO TEST IF M8200, OR M8204
5588  021760  104432                        TRAP    C$EXIT
5589  021762  000336                        .WORD   L10075-.
5590  021764                                MSTCLR
5591
5592  021770  012737  000000  002406        MOV     #0, FLAG                ;FLAG TO REPRESENT
5593                                                                        ;FIELD,PAGE
5594                                                                        ;TO VARIE STARTING PAGE,FIELD,
5595                                                                        ;CHANGE #0 PORTION OF INSTR.
5596  021776  012702  000000                MOV     #0, R2                  ;R2 TO CONTAIN JUMPED
5597                                                                        ;TO CHANGE STARTING IMM ADDR.,
```

```
5598
5599  022002  012737  000000  002412          MOV     #0, FADR        ;VARIE #0 PORTIONS OF INSTR.
5600                                          ;LOOP HERE              ;ADDRESS
5601  022010                          1$:
5602  022010  042737  000017  022050          BIC #17,2$              ;CLEAR JUNK FROM FIELD
5603                                                                  ;PORTION OF CHANGE FIELD INSTR
5604  022016  013700  002406                  MOV FLAG,R0             ;INORDER TO INC, DEC FIELD,PAGE
5605  022022  042700  177760                  BIC #^C<17>,R0
5606  022026  050037  022050                  BIS R0,2$               ;NOW POSITION IN INSTR.
5607  022032  042737  077777  022064          BIC #077777,3$                  ;NOW FOR IMMED. BR INSTR.
5608  022040  050237  022064                  BIS R2,3$               ;NOW ADD IMMEDIATE ADDR
5609
5610
5611
5612  022044                                  ROMCLK
5613  022044  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
5614  022050  000400                  2$:     000400                  ;MOVE PAGE,FIELD # TO BREG.
5615  022052                                  ROMCLK
5616  022052  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
5617  022056  061233                          61233                   ;MOV    BREG TO PC HIGH REG.
5618  022060                                  SROMCLK
5619  022060  004537  003100                  JSR     R5,..SROMCLK
5620  022064  100000                  3$:     100000                  ;NOW CLOCK IT IN BY JMP FIELD INSTR.
5621
5622  022066                                  ROMCLK                  ;READ PC REG HI
5623  022066  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
5624  022072  121265                          121265
5625  022074                                  ROMCLK                  ;READ PC REG LOW
5626  022074  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
5627  022100  121244                          121244
5628
5629  022102  016104  000004                  MOV 4(R1),R4            ;READ PC REG (NOW IN SEL 4)
5630  022106  042704  170000                  BIC     #170000,R4      ;STRIP FOR ONLY PAGE,FIELD BITS.
5631
5632  022112  013705  022050                  MOV 2$,R5               ;NOW FROM ADDR WE WANTED TO
5633  022116  000305                          SWAB R5                 ;JUMP TO
5634  022120  042705  170377                  BIC #170377,R5          ;CLEAR JUNK
5635  022124  050205                          BIS R2,R5               ;ADD IMMED ADDR
5636  022126                                  SKIP06  5$
5637                                          ;GOTO 5$ IF M8206
5638  022136  105205                          INCB    R5              ;UPDATE ADDR. EXPECTED SENCE THE READ
5639  022140                          5$:
5640                                                                  ;OF THE IBUS <13> INC THE PC.
5641
5642  022140  020504                          CMP R5,R4               ;JUMP GO OK?
5643  022142  001406                          BEQ 4$                  ;YEA, CONTINUES
5644  022144                                  ERROR 15                ;FAILED TO JUMP PROPERLY.
5645  022150  104455                          TRAP    C$ERDF
5646  022152  000017                          .WORD   15
5647  022154  005055                          .WORD   EM0
5648  022156  010006                          .WORD   ERR15
5649                                                                  ;"FROM ADDR" REPRESENTS
5650                                                                  ;THE ADDRESS WE STARTED AT
5651                                                                  ;"TO ADDR" REPRESENTS WHERE
5652                                                                  ;WE EXPECTED TO JUMP TO.
5653                                                                  ;"BAD ADDR" REPRESENTS WHERE
```

```
                                                ;WE WENT TO.

                                .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
```

```
5710  022254                                ROMCLK                  ;REG PC REG HI, PUT IN PORT5
5711  022254  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
5712  022260  121265                        121265
5713
5714  022262                                ROMCLK                  ;REG PC REG LOW, PUT IN PORT4
5715  022262  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
5716  022266  121244                        121244
5717  022270  005005                        CLR     R5              ;EXPECT ZERO
5718  022272  016104  000004                MOV     4(R1),R4        ;READ PC REG FROM PORT 4&5
5719  022276  042704  170003                BIC     #170003,R4
5720  022302  001406                        BEQ     40$             ;IF CLEARED, EXIT
5721                                                                ;NOTE WE ALSO CLEARED BIT 1 OF THE
5722                                                                ;PC REG, BECAUSE AFTER THE MASTER
5723                                                                ;CLEAR, WE DID TWO INSTRUCTIONS TO
5724                                                                ;READ IT, THUS CAUSING THE PC REG
5725                                                                ;TO GET BUMPED.
5726
5727  022304                                ERROR   45              ;MASTER CLEAR FAILED TO CLEAR
5728  022310  104455                        TRAP    C$ERDF
5729  022312  000055                        .WORD   45
5730  022314  005055                        .WORD   EMO
5731  022316  011124                        .WORD   ERR45
5732                                                                ;PC REG
5733  022320                        40$:
5734  022320                        ENDTST
5735  022320                        L10075:
5736  022320  104401                        TRAP    C$ETST
5737  022322                                BADHEAD
5738                                        ;************** TEST 28 **************
5739                                        ;*
5740                                        ;*JUMP TEST, JUMP ALWAYS, JUMP CHANGE FIELD
5741                                        ;*
5742                                        ;*IN THIS TEST, WE WILL CHECK THE ABILITY OF THE
5743                                        ;*MICRO PROCESSOR TO JUMP (BRANCH & ALWAYS INSTRUCTION)
5744                                        ;*TO LOCATIONS, FIELDS FROM OTHER LOCATIONS FIELDS.
5745                                        ;*WE ALREADY KNOW THAT THE BRANCH INSTR WORKS FROM
5746                                        ;*OTHER TEST.                PROCEDURE:
5747                                        ;*      1. START ADDR 0, FIELD 0
5748                                        ;*      2. **CALCULATE NEW ADDRS, FIELD  VIA DEC,
5749                                        ;*      3. CAUSE JUMP (BRANCH) TO NEW ADDRESS
5750                                        ;*      4. READ PC FROM IBUS*12 AND IBUS*13
5751                                        ;*      5. REPEAT STEP 2-4 256.TIMES
5752                                        ;*
5753                                        ;*      TO CALCULATE NEW ADDRESS:
5754                                        ;*      1. DEC LOW BYTE OF ADDRESS FOR PC ADDRES 0-7
5755                                        ;*      2. DEC LOW BYTE OF NADDRESS FOR PC ADDRESS 8-11
5756                                        ;*         BITS REPRESENTED AS BITS 0-3. WHEN  0-3 OVERFLOWS,
5757                                        ;*         RESTARTS AT ZERO.
5758                                        ;*         NET RESULT IS JUMPS FROM:
5759                                        ;*              FIELD,PAGE              LOC
5760                                        ;*                   0                  0
5761                                        ;*                   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)
```

```
5818  022464  042704  170000          BIC     #170000,R4      ;STRIP FOR ONLY PAGE,FIELD BITS.
5819
5820  022470  013705  022426          MOV 2$,R5               ;NOW FROM ADDR WE WANTED TO
5821  022474  000305                  SWAB R5                 ;JUMP TO
5822  022476  042705  170377          BIC #170377,R5          ;CLEAR JUNK
5823  022502  050205                  BIS R2,R5               ;ADD IMMED ADDR
5824  022504                          SKIP06  5$
5825                                  ;GOTO 5$ IF M8206
5826  022514  105205                  INCB    R5              ;UPDATE ADDR. EXPECTED SENCE THE READ
5827  022516                  5$:
5828                                                          ;OF THE IBUS <13> INC THE PC.
5829
5830  022516  020504                  CMP R5,R4               ;JUMP GO OK?
5831  022520  001406                  BEQ 4$                  ;YEA, CONTINUES
5832  022522                          ERROR 15                ;FAILED TO JUMP PROPERLY.
5833  022526  104455                  TRAP    C$ERDF
5834  022530  000017                  .WORD   15
5835  022532  005055                  .WORD   EM0
5836  022534  010006                  .WORD   ERR15
5837                                                          ;"FROM ADDR" REPRESENTS
5838                                                          ;THE ADDRESS WE STARTED AT
5839                                                          ;"TO ADDR" REPRESENTS WHERE
5840                                                          ;WE EXPECTED TO JUMP TO,
5841                                                          ;"BAD ADDR" REPRESENTS WHERE
5842                                                          ;WE WENT TO.
5843
5844                          .REM %
5845                                          15      12      11      10      9 8 7           0
5846                                          ----------------------------------------------
5847                                          !UNUSED !  FIELD        ! PAGE    !IMMED. ADDR  !
5848                                          !       !   BITS        ! BITS    !             !
5849                                          ----------------------------------------------
5850
5851                                  ;THIS IS A PICTURE OF THE P.C. REG.
5852                                   BITS 0-7 ARE IN IBUS*<12>
5853                                   BITS 8-11 ARE IN IBUS*<13>
5854                                   THEY GOT CLOCK IN THERE VIA JUMPS TAKEN
5855                                   THE FIELD BITS
5856                                   ARE IN BIT POSITION 0,1 OF THE INSTRUCTION AT 2$.
5857
5858                                   3$ WAS THE JUMP ALWAYS INSTRUCTION. THE IMMED. ADDR.
5859                                   WAS IN 0-7 OF THE JUMP INSTR.  THE PAGE BITS,
5860                                   PC REG BITS 8,9, WERE IN BITS 11,12 OF THE INSTR.
5861                                   JUMP INSTRUCTIONS HAVE BEEN CHECKED OUT
5862                                   BEFORE, SO THE IMPORTANT THING TO REMEMBER TO
5863                                   WATCH IS THE "FROM ADDR","TO ADDR"
5864
5865                          %
5866
5867  022536                  4$:     ESCAPE TST
5868  022536  104410                  TRAP    C$ESCAPE
5869  022540  000016                  .WORD   L10076-.
5870  022542  010437  002412          MOV R4,FADR
5871  022546  005337  002406          DEC FLAG                ;UPDATE PAGE,FIELD
5872  022552  105302                  DECB R2                 ;UPDATE JMMED. ADDR
5873  022554  001304                  BNE 1$                  ;LOOP IF NOT DONE.
```

```
5874
5875
5876  022556                          ENDTST
5877  022556                          L10076:
5878  022556  104401                          TRAP    C$ETST
5879  022560                                  BADHEAD
5880                                          ;************** TEST 29 **************
5881                                          ;*
5882                                          ;* IN THIS TEST WE'LL VERIFY THAT THE Z BIT CAN BE READ FROM
5883                                          ;* IBUS*<13>. WE ALLREADY KNOW THAT THE Z BIT WORKS PROPERLY,
5884                                          ;* ALL WE WANT TO KNOW HERE IS THAT IT CAN BE READ.
5885                                          ;*
5886  022560                                  BADHEAD
5887                                          ;************** TEST 29 **************
5888
5889  022560                                  BGNTST
5890  022560                          T29::
5891  022560                                  K4ONLY                          ;M8206 &M8207 ONLY!
5892                                          ;DO NOT DO TEST IF M8200, OR M8204
5893  022570  104432                          TRAP    C$EXIT
5894  022572  000200                          .WORD   L10077-.
5895  022574                                  MSTCLR
5896  022600                                  MYINT
5897  022600  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
5898  022604  004737  003166                  JSR     PC,CLRALL               ;CLR CONDITION CODES.
5899  022610                                  ROMCLK                          ;NOW READ IBUS*<15>PUT IN PORT 4
5900  022610  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
5901  022614  121264                          121264
5902  022616  116104  000004                  MOVB    4(R1),R4                ;READ IT FROM PORT 4
5903  022622  042704  177477                  BIC     #177477,R4              ;STRIP ANY JUNK,C&Z BITS 6,7
5904  022626  012705  000000                  MOV     #0,R5                   ;EXPECT IT CLEAR
5905  022632  120405                          CMPB    R4,R5                   ;OK?
5906  022634  001410                          BEQ     1$
5907  022636                                  ERROR   16                      ;FAILURE OF Z&C TO BE CLEAR.
5908  022642  104455                          TRAP    C$ERDF
5909  022644  000020                          .WORD   16
5910  022646  005055                          .WORD   EM0
5911  022650  010134                          .WORD   ERR16
5912
5913  022652                                  ESCAPE TST
5914  022652  104410                          TRAP    C$ESCAPE
5915  022654  000116                          .WORD   L10077-.
5916  022656  004737  003312          1$:     JSR     PC,SETZ                 ;SET Z BIT.
5917  022662                                  ROMCLK                          ;NOW GO BACK AND CHECK Z BIT SET.
5918  022662  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
5919  022666  121264                          121264
5920
5921  022670  016104  000004                  MOV     4(R1),R4                ;GET INFO.
5922  022674  042704  177477                  BIC     #^C<300>,R4             ;STRIP FOR C&Z BITS.
5923  022700  012705  000200                  MOV     #200,R5                 ;EXPECT ONLY Z BIT SET.
5924  022704  120405                          CMPB    R4,R5                   ;SET OK?
5925  022706  001410                          BEQ     2$
5926  022710                                  ERROR   16                      ;Z BIT FAILED TO SET PROPERLY.
5927  022714  104455                          TRAP    C$ERDF
5928  022716  000020                          .WORD   16
5929  022720  005055                          .WORD   EM0
```

```
5930  022722  010134                          .WORD   ERR16
5931
5932  022724                                  ESCAPE  TST
5933  022724  104410                          TRAP    C$ESCAPE
5934  022726  000044                          .WORD   L10077-.
5935  022730  004737  003166          2$:     JSR     PC,CLRALL               ;NOW TRY TO CLEAR Z BIT.
5936  022734                                  ROMCLK
5937  022734  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
5938  022740  121264                          121264
5939  022742  016104  000004                  MOV     4(R1),R4
5940  022746  042704  177477                  BIC     #^C<300>,R4             ;STRIP FOR C&Z BITS
5941  022752  001407                          BEQ     3$                      ;IF ZERO,WE'RE OK
5942  022754  005005                          CLR     R5                      ;ELSE REPORT ERROR
5943  022756                                  ERROR   16                      ;Z BIT FAILED TO CLEAR PROPERLY.
5944  022762  104455                          TRAP    C$ERDF
5945  022764  000020                          .WORD   16
5946  022766  005055                          .WORD   EM0
5947  022770  010134                          .WORD   ERR16
5948  022772                          3$:
5949  022772                                  ENDTST
5950  022772                          L10077:
5951  022772  104401                          TRAP    C$ETST
5952                                  ;FINDFAST
5953  022774                                  BADHEAD
5954                                          ;************** TEST 30 **************
5955                                          ;*
5956                                          ;* IN THIS TEST WE'LL VERIFY THAT THE C BIT CAN BE READ FROM
5957                                          ;* IBUS*<13>. WE ALLREADY KNOW THAT THE C BIT WORKS PROPERLY,
5958                                          ;* ALL WE WANT TO KNOW HERE IS THAT IT CAN BE READ.
5959                                          ;*
5960  022774                                  BADHEAD
5961                                          ;************** TEST 30 **************
5962
5963  022774                                  BGNTST
5964  022774                          T30::
5965  022774                                  K4ONLY                          ;M8206 &M8207 ONLY!
5966                                          ;DO NOT DO TEST IF M8200, OR M8204
5967  023004  104432                          TRAP    C$EXIT
5968  023006  000200                          .WORD   L10100-.
5969  023010                                  MSTCLR
5970  023014                                  MYINT
5971  023014  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
5972  023020  004737  003166                  JSR     PC,CLRALL               ;CLR CONDITION CODES.
5973  023024                                  ROMCLK                          ;NOW READ IBUS*<13>PUT IN PORT 4
5974  023024  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
5975  023030  121264                          121264
5976  023032  116104  000004                  MOVB    4(R1),R4                ;READ IT FROM PORT 4
5977  023036  042704  177477                  BIC     #177477,R4              ;STRIP ANY JUNK,C&Z BITS 6,7
5978  023042  012705  000000                  MOV     #0,R5                   ;EXPECT IT CLEAR
5979  023046  120405                          CMPB    R4,R5                   ;OK?
5980  023050  001410                          BEQ     1$
5981  023052                                  ERROR   16                      ;FAILURE OF Z&C TO BE CLEAR.
5982  023056  104455                          TRAP    C$ERDF
5983  023060  000020                          .WORD   16
5984  023062  005055                          .WORD   EM0
5985  023064  010134                          .WORD   ERR16
```

```
5986
5987  023066                                ESCAPE TST
5988  023066  104410                        TRAP    C$ESCAPE
5989  023070  000116                        .WORD   L10100-.
5990  023072  004737  003260          1$:   JSR     PC,SETC                 ;SET C BIT.
5991  023076                                ROMCLK                          ;NOW GO BACK AND CHECK C BIT SET.
5992  023076  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
5993  023102  121264                        121264
5994  023104  016104  000004                MOV     4(R1),R4                ;GET INFO.
5995  023110  042704  177477                BIC     #^C<300>,R4             ;STRIP FOR C&Z BITS.
5996  023114  012705  000100                MOV     #100,R5                 ;EXPECT ONLY C BIT SET.
5997  023120  120405                        CMPB    R4,R5                   ;SET OK?
5998  023122  001410                        BEQ     2$
5999  023124                                ERROR   16                      ;C BIT FAILED TO SET PROPERLY.
6000  023130  104455                        TRAP    C$ERDF
6001  023132  000020                        .WORD   16
6002  023134  005055                        .WORD   EM0
6003  023136  010134                        .WORD   ERR16
6004
6005  023140                                ESCAPE  TST
6006  023140  104410                        TRAP    C$ESCAPE
6007  023142  000044                        .WORD   L10100-.
6008  023144  004737  003166          2$:   JSR     PC,CLRALL               ;NOW TRY TO CLEAR C BIT.
6009  023150                                ROMCLK
6010  023150  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6011  023154  121264                        121264
6012  023156  016104  000004                MOV     4(R1),R4
6013  023162  042704  177477                BIC     #^C<300>,R4             ;STRIP FOR C&Z BITS
6014  023166  001407                        BEQ     3$                      ;IF ZERO,WE'RE OK
6015  023170  005005                        CLR     R5                      ;ELSE REPORT ERROR
6016  023172                                ERROR   16                      ;C BIT FAILED TO CLEAR PROPERLY.
6017  023176  104455                        TRAP    C$ERDF
6018  023200  000020                        .WORD   16
6019  023202  005055                        .WORD   EM0
6020  023204  010134                        .WORD   ERR16
6021  023206                          3$:
6022  023206                                ENDTST
6023  023206                          L10100:
6024  023206  104401                        TRAP    C$ETST
6025  023210                                BADHEAD
6026                                        ;************** TEST 31 **************
6027                                        ;*TEST OF PROGRAM CLOCK BIT
6028                                        ;*DO A MASTER CLEAR, VERIFY THAT PROGRAM CLOCK IS SET
6029                                        ;*WRITE PROGRAM CLOCK BIT TO A ONE, VERIFY THAT IT CLEARS,
6030                                        ;*AND THEN SETS SOME TIME LATER
6031  023210                                BADHEAD
6032                                        ;************** TEST 31 **************
6033
6034  023210                          BGNTST
6035  023210                          T31::
6036  023210                                MYINT
6037  023210  013701  002516                MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
6038  023214                                MSTCLR                  ;MASTER CLEAR M8200,4,7
6039  023220  005037  002440                CLR     TEMP            ;PREPARE FOR
6040  023224  005037  002444                CLR     $TMP0           ;DELAY
6041  023230  012761  000020  000004  1$:   MOV     #20,4(R1)       ;LOAD PORT 4
```

```
6042  023236  152761  000002  000001          BISB    #BIT1,1(R1)      ;SET ROMI
6043  023244  012761  121111  000006          MOV     #121111,6(R1)    ;SEL6   INSTRUCTION
6044  023252  152761  000003  000001          BISB    #BIT1!BIT0,1(R1);SET CLOCK BIT
6045  023260  012761  121224  000006          MOV     #121224,6(R1)    ;LOAD NEXT INSTRUCTION
6046  023266  152761  000003  000001          BISB    #BIT1!BIT0,1(R1);READ CLOCK BIT
6047  023274  142761  030001  000001          BICB    #BIT!BIT0,1(R1);CLEAR MAINT BITS
6048  023302  016104  000004                  MOV     4(R1),R4         ;PUT "FOUND" IN R4
6049  023306  005037  002452                  CLR     $GDDAT           ;PUT "EXPECTED" IN $GDDAT
6050  023312  123704  002452                  CMPB    $GDDAT,R4                ;IS PGM CLOCK CLEAR?
6051  023316  001406                          BEQ     2$
6052  023320  013702  002452                  MOV     $GDDAT,R2
6053  023324                                  ERRDF   50,EMB50                 ;ERROR, PGM CLOCK IS NOT CLEAR
6054  023324  104455                          TRAP    C$ERDF
6055  023326  000062                          .WORD   50
6056  023330  005545                          .WORD   EMB50
6057  023332  000000                          .WORD   0
6058  023334                          2$:
6059  023334                                  ROMCLK                   ;NEXT WORD IS INSTRUCTION,
6060  023334  004537  003044                  JSR     R5,..ROMCLK      ;CLOCK INSTRUCTION
6061  023340  121224                          121224                   ;PORT4 LU11
6062  023342  122761  000020  000004          CMPB    #20,4(R1)        ;IS PGM CLOCK SET?
6063  023350  001420                          BEQ     3$               ;BR IF YES
6064  023352  005237  002440                  INC     TEMP             ;INCREMENT DELAY
6065  023356  005537  002444                  ADC     $TMP0            ;INCREMENT DELAY
6066  023362  022737  000006  002444          CMP     #6,$TMP0         ;IS DELAY DONE
6067  023370  001361                          BNE     2$               ;BR IF NO
6068  023372  012702  000006                  MOV     #6,R2
6069  023376  013704  002444                  MOV     $TMP0,R4
6070  023402                                  ERRDF   51,EMB51                 ;ERROR PGM CLOCK NOT SET
6071  023402  104455                          TRAP    C$ERDF
6072  023404  000063                          .WORD   51
6073  023406  005577                          .WORD   EMB51
6074  023410  000000                          .WORD   0
6075  023412                          3$:
6076
6077  023412  122737  000007  002414          CMPB    #7,WTYPE         ; ONLY DO NEXT TEST IF M8207
6078  023420  001013                          BNE     4$               ; EXIT IF NOT.
6079
6080  023422  005737  002444                  TST     $TMP0            ; IF ANY LARGE COUNT, WE'RE OK
6081  023426  001010                          BNE     4$               ; THEN EXIT
6082
6083  023430  042737  000007  002440          BIC     #7,TEMP          ; CLEAR OUT ANY SMALL COUNT
6084  023436  001004                          BNE     4$               ; IF LARGE COUNT LEFT OVER, WE'RE OK.
6085
6086  023440                                  ERRDF   100,EMB1         ; ERROR
6087  023440  104455                          TRAP    C$ERDF
6088  023442  000144                          .WORD   100
6089  023444  005410                          .WORD   EMB1
6090  023446  000000                          .WORD   0
6091                                                                   ; TIME TTOOO SHORT FOR CLOCK.  MUST BE
6092                                                                   ; DEFECTIVE CAPACITOR IN TIMEING CIRCUIT.
6093  023450                          4$:
6094
6095  023450                          ENDTST
6096  023450                          L10101:
6097  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$RESET
023454                                  MYINT
023454  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
                                                                ;R1 CONTAINS BASE M8200,4,7 ADDRESS
023460                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
023464  005037  002440                  CLR     TEMP            ;PREPARE FOR DELAY
023470  013737  000024  002444          MOV     @#24,$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
```

```
6154  023660  005055                          .WORD   EM0
6155  023662  010256                          .WORD   ERR17
6156  023664  013737  002444  000024          MOV     $TMP0,@#24      ;RESTORE TRUE POWER FAIL ADDRESS
6157  023672  013706  002344                  MOV     PSTACK,SP       ;RESTORE STACK
6158  023676  032711  004000          4$:     BIT     #BIT11,(R1)     ;BIT11 STILL SET?
6159  023702  001016                          BNE     7$
6160  023704  005737  002470                  TST     RUNB
6161  023710  001013                          BNE     7$
6162  023712  011104                          MOV     (R1),R4
6163  023714  012705  004000                  MOV     #BIT11,R5
6164  023720                                  ERROR   35              ;OAC FAILED
6165  023724  104455                          TRAP    C$ERDF
6166  023726  000043                          .WORD   35
6167  023730  005055                          .WORD   EM0
6168  023732  010472                          .WORD   ERR35
6169                                                                  ;TO PREVENT
6170                                                                  ;INIT FROM
6171                                                                  ;CLEARING CSR
6172  023734                          EXIT    TST
6173  023734  104432                          TRAP    C$EXIT
6174  023736  000104                          .WORD   L10102-.
6175  023740  012711  003000          7$:     MOV     #BIT9!BIT10,(R1) ;SEL6 = MAINT IR
6176  023744  012705  121111                  MOV     #121111,R5      ;R5 = EXPECTED
6177  023750  016104  000006                  MOV     6(R1),R4        ;R4 = FOUND
6178  023754  020504                          CMP     R5,R4           ;MAINT IR SHOULD = 12111
6179  023756  001431                          BEQ     6$              ;BR IF OK
6180  023760                                  MSTCLR
6181  023764                                  ERROR   35              ;IF = 0 THEN BUS INIT WAS
6182  023770  104455                          TRAP    C$ERDF
6183  023772  000043                          .WORD   35
6184  023774  005055                          .WORD   EM0
6185  023776  010472                          .WORD   ERR35
6186                                                                  ;NOT BLOCKED FROM CLEARING
6187                                                                  ;THE M8200,4,7
6188
6189  024000  000000          11$:    .WORD   0                       ;TEMP COUNT FOR STALL ON POWER UP.
6190
6191  024002  052711  040000          10$:    BIS     #BIT14,(R1)     ;CLR THE THING SO IT CAN'T ASSIRT AC LOW
6192                                                                  ;AGAIN!
6193  024006                                  MSTCLR
6194  024012                                  ERROR   17              ;ERROR  GLIP GAVE US SECOUND UNEXPECTED
6195  024016  104455                          TRAP    C$ERDF
6196  024020  000021                          .WORD   17
6197  024022  005055                          .WORD   EM0
6198  024024  010256                          .WORD   ERR17
6199                                                                  ;ASSERTION OF AC LOW ON UNIBUS.
6200                                                                  ;FATEL TYPE OF ERROR.
6201  024026  062706  000004                  ADD     #4,SP           ;RESTORE STACK.
6202  024032  012637  000024                  MOV     (SP)+,@#24
6203  024036                                  MSTCLR
6204  024042                          6$:
6205  024042                          ENDTST
6206  024042                          L10102:
6207  024042  104401                          TRAP    C$ETST
6208
6209  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,
```

```
6266  024272  004537  003044                 JSR     R5,..ROMCLK      ;CLOCK INSTRUCTION
6267  024276  123100                 8$:     123100                   ;WRITE ZERO TO SP
6268  024300  112761  000377  000004         MOVB    #377,4(R1)       ;LOAD PORT4
6269  024306                                 ROMCLK                   ;NEXT WORD IS INSTRUCTION,
6270  024306  004537  003044                 JSR     R5,..ROMCLK      ;CLOCK INSTRUCTION
6271  024312  123100                 9$:     123100                   ;WRITE ALL ONES TO SP
6272  024314  110261  000004                 MOVB    R2,4(R1)         ;LOAD PORT4
6273  024320                                 ROMCLK                   ;NEXT WORD IS INSTRUCTION,
6274  024320  004537  003044                 JSR     R5,..ROMCLK      ;CLOCK INSTRUCTION
6275  024324  123100                 10$:    123100                   ;WRITE SP ADDRESS TO ITSELF
6276  024326  005202                         INC     R2               ;NEXT SP ADDRESS
6277  024330  022702  000020                 CMP     #20,R2           ;DONE YET?
6278  024334  001335                         BNE     7$               ;BR IF NO
6279  024336  005002                         CLR     R2               ;R2 = ,AOM ,E, ADDRESS
6280  024340                                 ROMCLK                   ;NEXT WORD IS INSTRUCTION,
6281  024340  004537  003044                 JSR     R5,..ROMCLK      ;CLOCK INSTRUCTION
6282  024344  010000                         010000                   ;MAR _ 0
6283  024346                                 ROMCLK
6284  024346  004537  003044                 JSR     R5,..ROMCLK      ;CLOCK INSTRUCTION
6285  024352  004000                         4000
6286  024354  105061  000004         11$:    CLRB    4(R1)            ;CLEAR PORT4
6287  024360                                 ROMCLK                   ;NEXT WORD IS INSTRUCTION,
6288  024360  004537  003044                 JSR     R5,..ROMCLK      ;CLOCK INSTRUCTION
6289  024364  122500                         122500                   ;WRITE ZEROS TO MEM
6290  024366  112761  000377  000004         MOVB    #377,4(R1)       ;LOAD PORT4
6291  024374                                 ROMCLK                   ;NEXT WORD IS INSTRUCTION,
6292  024374  004537  003044                 JSR     R5,..ROMCLK      ;CLOCK INSTRUCTION
6293  024400  122500                         122500                   ;WRITE ONES TO MEM
6294  024402  110261  000004                 MOVB    R2,4(R1)         ;LOAD PORT4
6295  024406                                 ROMCLK                   ;NEXT WORD IS INSTRUCTION,
6296  024406  004537  003044                 JSR     R5,..ROMCLK      ;CLOCK INSTRUCTION
6297  024412  136500                         136500                   ;WRITE TO MEM IT OWN ADDRESS
6298  024414  005202                         INC     R2               ;NEXT MEM ADDRESS
6299  024416  022702  001000                 CMP     #1000,R2         ;DONE YET?
6300  024422  001354                         BNE     11$              ;BR IF NO
6301
6302                                         ;NOW GO BACK AND READ EVERYTHIN
6303
6304  024424                                 ROMCLK                   ;NEXT WORD IS INSTRUCTION,
6305  024424  004537  003044                 JSR     R5,..ROMCLK      ;CLOCK INSTRUCTION
6306  024430  010000                         010000                   ;MAR 0
6307  024432                                 ROMCLK                   ;NEXT WORD IS INSTRUCTION,
6308  024432  004537  003044                 JSR     R5,..ROMCLK      ;CLOCK INSTRUCTION
6309  024436  004000                         4000                     ;MAR HI _ 0 (M8200,4,7 ONLY)
6310                                         ;WOULD BE CRAM CODE
6311  024440  005737  002432                 TST     TYPE
6312  024444  001452                         BEQ     40$
6313  024446  005005                         CLR     R5               ;R5 IS INDEX REGISTER
6314  024450  042737  000360  024512  12$:   BIC     #360,13$         ;CLEAR ADDRESS FIELD
6315  024456  116502  025100                 MOVB    30$(R5),R2       ;R2 = IBUS* ADDRESS
6316  024462  010203                         MOV     R2,R3            ;PUT IBUS* ADDRESS IN R3
6317  024464  006303                         ASL     R3               ;SHIFT ADDRESS TO BITS 4-7
6318  024466  006303                         ASL     R3
6319  024470  006303                         ASL     R3
6320  024472  006303                         ASL     R3
6321  024474  050337  024512                 BIS     R3,13$           ;ADD ADDRESS TO INSTRUCTION
```

```
6322  024500  116537  025106  002452          MOVB    31$(R5),$GDDAT  ;$GDDAT = "EXPECTED"
6323  024506                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6324  024506  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6325  024512  121004                  13$:    121004                  ;PORT4 - IBUS* REGISTER
6326  024514  016104  000004                  MOV     4(R1),R4        ;R4 = "FOUND"
6327  024520  123704  002452                  CMPB    $GDDAT,R4               ;IBUS* CONTENTS OK?
6328  024524  001416                          BEQ     20$             ;BR IF YES
6329  024526  010237  002434                  MOV     R2,MR0
6330  024532  105037  002453                  CLRB    $GDDAT+1
6331  024536  013705  002452                  MOV     $GDDAT,R5
6332  024542                                  ERROR   29              ;IBUS* DATA ERROR
6333  024546  104455                          TRAP    C$ERDF
6334  024550  000035                          .WORD   29
6335  024552  005055                          .WORD   EM0
6336  024554  010350                          .WORD   ERR29
6337  024556                                  ESCAPE  TST
6338  024556  104410                          TRAP    C$ESCAPE
6339  024560  000334                          .WORD   L10103-.
6340  024562  005205                  20$:    INC     R5              ;INC COUNTER
6341  024564  022705  000005                  CMP     #5,R5           ;DONE YET?
6342  024570  001327                          BNE     12$             ;BR IF NO
6343
6344  024572                          40$:
6345                                  ;END CRAM,GENERAL TESTS
6346
6347  024572  005002                          CLR     R2              ;R2 = IBUS REG ADDRESS
6348  024574  042737  000360  024630  14$:    BIC     #360,15$        ;CLEAR ADDRESS FIELD OF INSTRUCTION
6349  024602  010203                          MOV     R2,R3           ;R3 = IBUS ADDRESS
6350  024604  006303                          ASL     R3              ;SHIFT ADDRESS TO BITS 4-7
6351  024606  006303                          ASL     R3
6352  024610  006303                          ASL     R3
6353  024612  006303                          ASL     R3
6354  024614  050337  024630                  BIS     R3,15$          ;ADD ADDRESS TO INSTRUCTION
6355  024620  010237  002452                  MOV     R2,$GDDAT               ;$GDDAT = "EXPECTED"
6356  024624                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6357  024624  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6358  024630  021004                  15$:    021004                  ;PORT4 - IBUS REG
6359  024632  016104  000004                  MOV     4(R1),R4        ;IBUS = "FOUND"
6360  024636  123704  002452                  CMPB    $GDDAT,R4               ;IBUS CONTENTS OK?
6361  024642  001410                          BEQ     21$             ;BR IF YES
6362  024644  013705  002452                  MOV     $GDDAT,R5
6363  024650                                  ERROR   29              ;IBUS DATA ERROR
6364  024654  104455                          TRAP    C$ERDF
6365  024656  000035                          .WORD   29
6366  024660  005055                          .WORD   EM0
6367  024662  010350                          .WORD   ERR29
6368  024664  005202                  21$:    INC     R2              ;NEXT IBUS REGISTER
6369  024666  022702  000010                  CMP     #10,R2          ;DONE YET?
6370  024672  001340                          BNE     14$             ;BR IF NO
6371  024674  005002                          CLR     R2              ;R2 = SP ADDRESS
6372  024676  042737  000017  024714  16$:    BIC     #17,17$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
6373  024704  050237  024714                  BIS     R2,17$          ;ADD ADDRESS TO INSTRUCTION
6374  024710                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6375  024710  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6376  024714  040600                  17$:    040600                  ;BR - SP
6377  024716  010237  002452                  MOV     R2,$GDDAT               ;$GDDAT = "EXPECTED"
```

```
6378  024722                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
6379  024722  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6380  024726  061224                          061224                  ;PORT4   BR
6381  024730  016104  000004                  MOV     4(R1),R4        ;R4 = 'FOUND'
6382  024734  123704  002452                  CMPB    $GDDAT,R4               ;SP CONTENTS OK?
6383  024740  001412                          BEQ     22$             ;BR IF YES
6384  024742  013705  002452                  MOV     $GDDAT,R5
6385  024746                                  ERROR   7               ;SP DATA ERROR
6386  024752  104455                          TRAP    C$ERDF
6387  024754  000007                          .WORD   7
6388  024756  005055                          .WORD   EMO
6389  024760  007026                          .WORD   ERR7
6390  024762                                  ESCAPE  TST
6391  024762  104410                          TRAP    C$ESCAPE
6392  024764  000130                          .WORD   L10103-.
6393  024766  005202              22$:        INC     R2              ;NEXT SP LOCATION
6394  024770  022702  000020                  CMP     #20,R2          ;DONE YET?
6395  024774  001340                          BNE     16$             ;BR IF NO
6396  024776  005002                          CLR     R2              ;R2 = MEMORY ADDRESS
6397  025000                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6398  025000  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6399  025004  010000                          010000                  ;MAR   0
6400  025006                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6401  025006  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6402  025012  004000                          4000                    ;MAR HI   0 (M8200,4,7 OR FAMILY ONLY)
6403  025014  010237  002452          18$:    MOV     R2,$GDDAT               ;$GDDAT = 'EXPECTED'
6404  025020                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6405  025020  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6406  025024  055224                          055224                  ;PORT4   MAIN MEM
6407  025026  016104  000004                  MOV     4(R1),R4        ;R4 = 'FOUND'
6408  025032  123704  002452                  CMPB    $GDDAT,R4               ;MAIN MEM CONTENTS OK?
6409  025036  001412                          BEQ     23$             ;BR IF YES
6410  025040  013705  002452                  MOV     $GDDAT,R5
6411  025044                                  ERROR   6               ;MAIN MEM DATA ERROR
6412  025050  104455                          TRAP    C$ERDF
6413  025052  000006                          .WORD   6
6414  025054  005055                          .WORD   EMO
6415  025056  006700                          .WORD   ERR6
6416  025060                                  ESCAPE  TST
6417  025060  104410                          TRAP    C$ESCAPE
6418  025062  000032                          .WORD   L10103-.
6419  025064  005202              23$:        INC     R2              ;NEXT MEM ADDRESS
6420  025066  022702  001000                  CMP     #1000,R2                ;DONE YET?
6421  025072  001350                          BNE     18$             ;BR IF NO
6422  025074                                  EXIT    TST
6423  025074  104432                          TRAP    C$EXIT
6424  025076  000016                          .WORD   L10103-.
6425  025100     000     002     003  30$:    .BYTE   0,2,3,5,10
6426  025103     005     010
6427
6428          025106                  .EVEN
6429  025106     001     003     004  31$:    .BYTE   1,3,4,6,10
6430  025111     006     010
```

```
                025114                  .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   EMO
```

```
                        .WORD   ERR7

                10$:
                        ENDTST
                L10104:
                        TRAP    C$ETST

                        BADHEAD
                        ;************** TEST 35 **************
                        ;*
                        ;* EXTENDED CRAM TEST FOR M8206. IN THIS TEST WE WILL LOAD DATA
                        ;* THROUGHOUT THE CRAM (TEST DATA IS JUST 4K OF DIAG. CODE) AND
                        ;* THEN READ IT BACK AND VERIFY THAT IT IS CORRECT.
                        BADHEAD
                        ;************** TEST 35 **************

                        BGNTST
                T35::
                        SKIP06  10$                     ;DO TEST ONLY IF IT IS A M8206
                        ;GOTO 10$ IF M8206
                        EXIT    TST                     ;OTHERWIZE,SKIP TEST.
                        TRAP    C.EXIT
                        .WORD   L10105-.

                10$:    MYINT
                        MOV     KMCSR,R1                ;RECORD DEVICE ADDR.

                        MOV     #ROMMAP,R2              ;GET ADDR. OF LIST.

                        MOV     #2000,(R1)              ;SET TO WRITE DATA.
                        CLR     R3                      ;CRAM ADDR ZERO.

                15$:    MOV     R3,4(R1)                ;SET ADDR.
                        MOV     (R2)+,6(R1)             ;WRITE DATA.

                        CMP     R3,MEMSZ                ;DONE WHOLE CRAM?
                        BEQ     20$                     ;YES,EXIT THIS LOOP.
                        INC     R3                      ;NO,UPDAT ADDR.
                        BR      15$
                20$:    CLR     R3                      ;NOW WE WILL READ BACK,STARTING AT
                                                        ;CRAM ADDR. ZERO.
                        MOV     #ROMMAP,R5              ;GET ADDR. LIST OF DATA

                30$:    MOV     R3,4(R1)                ;SET ADDR.

                        MOV     (R5),R2                 ;PUT EXPECTED INTO R2
                        MOV     6(R1),R4                ;READ ACCUAL
                        CMP     R2,R4                   ;EQUAL?
                        BEQ     40$                     ;YES,CONTINUE.
                        MOV     R3,R0

                        ERROR   1                       ;ERROR CRAM DATA TEST,DATA
                        TRAP    C$ERDF
                        .WORD   1
                        .WORD   EM0
                        .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
```

```
6599  025436  061220                          61220                           ;MOVE BRG,BSEL0
6600  025440  061222                          61222                           ;MOVE BRG,BSEL2
6601  025442  061223                          61223                           ;MOVE BRG,BSEL3
6602  025444  061224                          61224                           ;MOVE BRG,BSEL4
6603  025446  061225                          61225                           ;MOVE BRG,BSEL5
6604  025450  061226                          61226                           ;MOVE BRG,,BSEL6
6605  025452  061227                          61227                           ;MOVE BRG,BSEL7
6606  025454  123000                          123000                          ;MOVE BSEL0,SP0
6607  025456  104022                          104022                          ;BRANCH BACK ONE LOCATION.
6608  025460  177777                          177777
6609
6610  025462  012711  040000                  MOV     #040000,(R1)            ;INITIALIZE MCPU
6611  025466  012711  100000                  MOV     #100000,(R1)            ;START CPU.
6612
6613  025472  012700  000062                  MOV     #50.,R0                 ;THE CYCLE TIME ON THE M8206 IS
6614                                                                           ;200NS.  WE ARE ASKING THE MCPU TO
6615                                                                           ;DO 8 INSTRUCTIONS.  WE'LL DELAY
6616                                                                           ;100 PDP11 INSTRUCTIONS
6617                                                                           ;THIS REALLY SHOULD BE PLENTY OF TIME.
6618
6619  025476  005300                  20$:     DEC     R0
6620  025500  001376                          BNE     20$
6621
6622  025502  005005                          CLR     R5                      ;JUST FOR TYPEOUT.
6623  025504  012705  000377                  MOV     #377,R5         ;EXPECT 377
6624  025510  111104                          MOVB    (R1),R4         ;READ MCPU
6625  025512  120405                          CMPB    R4,R5           ;SEE IF OK.
6626  025514  001410                          BEQ     30$
6627
6628  025516                                  ERROR   29              ;ERROR! MCPU WAS TO WRITE ALL
6629  025522  104455                          TRAP    C$ERDF
6630  025524  000035                          .WORD   29
6631  025526  005055                          .WORD   EM0
6632  025530  010350                          .WORD   ERR29
6633                                                                           ;ONES INTO BSEL0,BUT INSTEAD FAILED.
6634  025532                                  ESCAPE  TST
6635  025532  104410                          TRAP    C$ESCAPE
6636  025534  000304                          .WORD   L10106-.
6637
6638  025536  012705  177777          30$:    MOV     #177777,R5              ;EXPECT ALL ONES
6639  025542  016104  000002                  MOV     2(R1),R4                ;RECIEVED
6640  025546  020405                          CMP     R4,R5                   ;RECIEVE OK?
6641  025550  001410                          BEQ     40$
6642
6643  025552                                  ERROR   29              ;ERROR! MCPU WAS TO WRITE ALL ONES
6644  025556  104455                          TRAP    C$ERDF
6645  025560  000035                          .WORD   29
6646  025562  005055                          .WORD   EM0
6647  025564  010350                          .WORD   ERR29
6648                                                                           ;INTO BSEL 2&3
6649
6650  025566                                  ESCAPE  TST
6651  025566  104410                          TRAP    C$ESCAPE
6652  025570  000250                          .WORD   L10106-.
6653
6654  025572  016104  000004          40$:    MOV     4(R1),R4                ;READ BSEL 4&5
```

```
6655  025576  020405                        CMP     R4,R5                   ;READ OK?
6656  025600  001410                        BEQ     50$
6657
6658  025602                                ERROR   29                      ;ERROR! FAILED TO WRITE BSEL $&5
6659  025606  104455                        TRAP    C$ERDF
6660  025610  000035                        .WORD   29
6661  025612  005055                        .WORD   EM0
6662  025614  010350                        .WORD   ERR29
6663                                                                        ; TO ALL ONES.
6664  025616                                ESCAPE  TST
6665  025616  104410                        TRAP    C$ESCAPE
6666  025620  000220                        .WORD   L10106-.
6667
6668  025622  016104  000006        50$:    MOV     6(R1),R4                ;READ BSEL 6&7
6669  025626  020405                        CMP     R4,R5                   ;READ OK?
6670  025630  001410                        BEQ     60$
6671
6672  025632                                ERROR   29                      ;ERROR! FAILED TO WRITE BSEL 6&7
6673  025636  104455                        TRAP    C$ERDF
6674  025640  000035                        .WORD   29
6675  025642  005055                        .WORD   EM0
6676  025644  010350                        .WORD   ERR29
6677                                                                        ; TO ALL ONES.
6678  025646                                ESCAPE  TST
6679  025646  104410                        TRAP    C$ESCAPE
6680  025650  000170                        .WORD   L10106-.
6681  025652  105011        60$:    CLRB    (R1)                            ;SIGNAL MCPU TO WRITE ALL ZEROS.
6682  025654  005005                        CLR     R5                      ;EXPECT TO READ ALL ZEROS.
6683
6684  025656  005004                        CLR     R4
6685  025660  111104                        MOVB    (R1),R4                 ;READ BSEL0
6686  025662  001410                        BEQ     70$                     ;EXPECT ZERO.
6687
6688  025664                                ERROR   29                      ;MCPU FAILED TO CLEAR BSEL0
6689  025670  104455                        TRAP    C$ERDF
6690  025672  000035                        .WORD   29
6691  025674  005055                        .WORD   EM0
6692  025676  010350                        .WORD   ERR29
6693
6694  025700                                ESCAPE  TST
6695  025700  104410                        TRAP    C$ESCAPE
6696  025702  000136                        .WORD   L10106-.
6697  025704  016104  000002        70$:    MOV     2(R1),R4                ;READ BSEL 2&3
6698  025710  001410                        BEQ     80$                     ;IF ZERO,OK
6699
6700  025712                                ERROR   29                      ;MCPU FAILED TO CLEAR BSEL 2&3
6701  025716  104455                        TRAP    C$ERDF
6702  025720  000035                        .WORD   29
6703  025722  005055                        .WORD   EM0
6704  025724  010350                        .WORD   ERR29
6705  025726                                ESCAPE  TST
6706  025726  104410                        TRAP    C$ESCAPE
6707  025730  000110                        .WORD   L10106-.
6708  025732                        80$:
6709  025732  016104  000004                MOV     4(R1),R4                ;READ BSEL 4&5
6710  025736  001410                        BEQ     90$
```

```
6711
6712  025740                                    ERROR   29                      ;MCPU FAILED TO CLEAR BSEL 4&5
6713  025744  104455                            TRAP    C$ERDF
6714  025746  000035                            .WORD   29
6715  025750  005055                            .WORD   EM0
6716  025752  010350                            .WORD   ERR29
6717  025754                                    ESCAPE  TST
6718  025754  104410                            TRAP    C$ESCAPE
6719  025756  000062                            .WORD   L10106-.
6720  025760                            90$:
6721  025760  016104  000006                    MOV     6(R1),R4                ;READ BSEL 6&7
6722  025764  001406                            BEQ     95$
6723
6724  025766                                    ERROR   29                      ;MCPU FAILED TO CLEAR BSEL 6&7
6725  025772  104455                            TRAP    C$ERDF
6726  025774  000035                            .WORD   29
6727  025776  005055                            .WORD   EM0
6728  026000  010350                            .WORD   ERR29
6729
6730  026002                            95$:    EXIT    TST
6731  026002  104432                            TRAP    C$EXIT
6732  026004  000034                            .WORD   L10106-.
6733
6734
6735                                            ;
6736                                            ;LOADER   SUBROUTINE USED BY THIS TEST TO LOAD MICRO CODE INTO A M8206
6737                                            ;
6738
6739  026006  012711  002000            LOADER: MOV     #2000,(R1)
6740
6741  026012  005000                            CLR     R0
6742
6743  026014  010061  000004            10$:    MOV     R0,4(R1)                ;SET ADDR.
6744  026020  005200                            INC     R0
6745  026022  011561  000006                    MOV     (R5),6(R1)              ;WRIITE MICRO CODE.
6746  026026  022527  177777                    CMP     (R5)+,#177777           ;SEE IF TERM.
6747  026032  001370                            BNE     10$
6748  026034  005011                            CLR     (R1)
6749  026036  000205                            RTS     R5
6750
6751  026040                                    ENDTST
6752  026040                            L10106:
6753  026040  104401                            TRAP    C$ETST
6754
6755  026042                                    BADHEAD
6756                                            ;************** TEST 37 **************
6757                                            ;*
6758                                            ;*NEGATIVE ADDRESS TEST.
6759                                            ;*      IN THIS TEST, WE'LL MAKE SURE THAT THE M8207
6760                                            ;*      DOES NOT RESPOND TO AN ADDRESS THAT ISN'T ASSIGNED
6761                                            ;*      TO IT
6762                                            ;*
6763  026042                                    BADHEAD
6764                                            ;************** TEST 37 **************
6765
6766  026042                                    BGNTST
```

```
6767  026042                                  T37::
6768  026042                                          MYINT
6769  026042  013701  002516                          MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
6770
6771  026046  012711  000641                          MOV     #641,(R1)       ;PUT A DEFINITE PATTERN IN MCPU.
6772  026052  012737  026130  000004                  MOV     #20$,@#4        ;SET UP FOR TRAPS FROM NON-EX.
6773  026060  005037  000006                          CLR     @#6
6774  026064  012702  160000                          MOV     #160000,R2      ;GET STARTING ADDRESS.
6775
6776  026070  022712  000641          10$:    CMP     -641,(R2)       ;SEE IF CONTENTS OF THE ADDRESS
6777                                                                  ;POINTED TO BY R2 EQUALS THE CONTENTS
6778                                                                  ;OF THE MCPU CSR
6779  026074  001420                          BEQ     40$
6780
6781  026076  062702  000002          15$:    ADD     #2,R2           ;UPDATE ADDRESS.
6782  026102  020227  177700                  CMP     R2,#177700      ;DONE?     ;80
6783  026106  001370                          BNE     10$             ;NO-LOOP
6784
6785  026110  013737  002464  000004  17$:    MOV     SAVE4,@#4       ;RESTORE TRAP CATCHER
6786  026116  013737  002466  000006          MOV     SAVE6,@#6       ;FROM VALUES SAVED BY INIT SECTION
6787  026124                                  EXIT    TST             ;EXIT, ALL DONE
6788  026124  104432                          TRAP    C$EXIT
6789  026126  000052                          .WORD   L10107-.
6790
6791  026130  062706  000004          20$:    ADD     #4,SP           ;SAVE FPOM TRAP
6792  026134  000760                          BR      15$             ;LOOP
6793
6794  026136                          40$:    ;*OH NO, WE MAY HAVE A DUAL ADDRESS PROBLEM!
6795
6796  026136  012711  000174                  MOV     #174,(R1)       ;WRITE NEW PATTERN IN MCPU CSR
6797  026142  022712  000174                  CMP     #174,(R2)       ;DID NEW PATTERN SHOW UP IN ADDR?
6798  026146  001403                          BEQ     60$
6799
6800  026150  012711  000641          50$:    MOV     #641,(R1)       ;PUT OLD PATTERN BACK IN MCPU CSR.
6801  026154  000750                          BR      15$             ;LOOP
6802
6803  026156  020102                  60$:    CMP     R1,R2           ;IS THIS THE MCPU ADDRESS?
6804  026160  001773                          BEQ     50$             ;YES-NO ERROR
6805
6806  026162                                  ERROR   40              ;DUAL ADDRESS ERROR
6807  026166  104455                          TRAP    C$ERDF
6808  026170  000050                          .WORD   40
6809  026172  005055                          .WORD   EM0
6810  026174  010636                          .WORD   ERR40
6811  026176  000744                          BR      17$
6812
6813  026200                                  ENDTST
6814  026200                          L10107:
6815  026200  104401                          TRAP    C$ETST
6816
6817  026202                                  BADHEAD
6818                                          ;************** TEST 38 **************
6819                                          ;*
6820                                          ;*BYTE ADDRESSING TEST
6821                                          ;*      HERE, WE'RE GOING TO MAKE SURE THAT WE CAN
6822                                          ;*      WRITE INTO ONLY A HIGH OR LOW BYTE OF THE MCPU.
```

```
6823
6824  026202                                    ;*
                                                BADHEAD
6825                                            ;************** TEST 38 **************
6826
6827  026202                                    BGNTST
6828  026202                              T38::
6829  026202                                    MYINT
6830  026202  013701  002516                    MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
6831  026206  005061  000002                    CLR     2(R1)           ;CLEAR CSR
6832  026212  112761  177777  000002            MOVB    #-1,2(R1)       ;WRITE ALL ONES INTO LOW BYTE
6833                                                                    ;OF CSR
6834  026220  032761  177400  000002            BIT     #177400,2(R1)   ;SEE IF HIGH BYTE GOT WRITTEN
6835  026226  001410                            BEQ     10$
6836
6837  026230                                    ERROR   41              ;HIGH BYTE GOT WRITTEN INTO ON A LOW BYTE
6838  026234  104455                            TRAP    C$ERDF
6839  026236  000051                            .WORD   41
6840  026240  005055                            .WORD   EMO
6841  026242  010702                            .WORD   ERR41
6842  026244                                    ESCAPE  TST             ;OPERATION
6843  026244  104410                            TRAP    C$ESCAPE
6844  026246  000040                            .WORD   L10110-.
6845
6846  026250  005061  000002            10$:    CLR     2(R1)
6847  026254  112761  177777  000003            MOVB    #-1,3(R1)       ;WRITE INTO HIGH BYTE
6848  026262  032761  000377  000002            BIT     #377,2(R1)      ;SEE IF LOW BYTE GOT WRITTEN
6849  026270  001406                            BEQ     20$
6850
6851  026272                                    ERROR   42              ;LOW BYTE GOT WRITTEN INTO ON A
6852  026276  104455                            TRAP    C$ERDF
6853  026300  000052                            .WORD   42
6854  026302  005055                            .WORD   EMO
6855  026304  010744                            .WORD   ERR42
6856                                                                    ;HIGH BYTE OPERATION.
6857
6858  026306                              20$:
6859  026306                                    ENDTST
6860  026306                              L10110:
6861  026306  104401                            TRAP    C$ETST
6862
6863  026310                                    BADHEAD
6864                                            ;************** TEST 39 **************
6865                                            ;*
6866                                            ;*IN THIS TEST WE'RE GOING TO MAKE SURE THAT THE PC
6867                                            ;*REG COUNTS UP PROPERLY. THE PC REG SHOULD INCREMENT
6868                                            ;*ONCE AFTER EACH INSTRUCTION.
6869                                            ;*
6870  026310                                    BADHEAD
6871                                            ;************** TEST 39 **************
6872
6873  026310                                    BGNTST
6874  026310                              T39::
6875  026310                                    SKIP07  10$             ;ONLY DO IF M8207
6876                                            ;GOTO 10$ IF M8207
6877  026320                                    EXIT TST
6878  026320  104432                            TRAP    C$EXIT
```

```
6879  026322  000122                        .WORD   L10111-.
6880
6881  026324                        10$:    MYINT
6882  026324  013701  002516                MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
6883  026330                                MSTCLR
6884  026334                                ROMCLK
6885  026334  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6886  026340  000400                        400
6887  026342                                ROMCLK
6888  026342  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6889  026346  061233                        61233
6890  026350                                SROMCLK
6891  026350  004537  003100                JSR     R5,..SROMCLK
6892  026354  100000                        100000
6893  026356  012705  000001                MOV     #1,R5           ;START AT ZERO
6894
6895  026362                        20$:    ROMCLK                  ;READ PC HIGH REG.
6896  026362  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6897  026366  121265                        121265
6898
6899  026370                                ROMCLK                  ;READ PC LOW REG.
6900  026370  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6901  026374  121244                        121244
6902  026376  016104  000004                MOV     4(R1),R4        ;GET WHOLE PICTURE
6903  026402  042704  170000                BIC     #170000,R4
6904  026406  020405                        CMP     R4,R5           ;INCREMENT OK?
6905  026410  001410                        BEQ     30$
6906
6907  026412                                ERROR   47              ;PC FAILED TO INCREMENT PROPERLY
6908  026416  104455                        TRAP    C$ERDF
6909  026420  000057                        .WORD   47
6910  026422  005055                        .WORD   EMO
6911  026424  011264                        .WORD   ERR47
6912                                                                ;SHOULD INCREMENT BY ONE
6913                                                                ;FOR EACH INSTRUCTION.
6914  026426                                ESCAPE  TST
6915  026426  104410                        TRAP    C$ESCAPE
6916  026430  000014                        .WORD   L10111-.
6917
6918  026432  062705  000002        30$:    ADD     #2,R5           ;UPDATE EXPECTED ADDRESS BY 2.
6919  026436  020527  000777                CMP     R5,#777
6920  026442  001347                        BNE     20$
6921
6922  026444                                ENDTST
6923  026444                        L10111:
6924  026444  104401                        TRAP    C$ETST
6925
6926  026446                                BADHEAD
6927                                        ;************** TEST 40 **************
6928                                        ;*
6929                                        ;*IN THIS TEST WE'LL MAKE SURE THAT 'BRANCH FIELD H'' DOESN'T
6930                                        ;*GET SUCH HIGH.
6931                                        ;*FIRST WE'LL CLEAR THE PC HIGH REG. THEN WE'LL DO A BRANCH INSTR
6932                                        ;*WITH BAB BITS 11&12 SET. IF PCR BITS 8&9 SET THEN WE'LL KNOW
6933                                        ;*WE WERE SUCCESSFUL IF PCR BITS 8&9 FAIL TO SET, WE'LL KNOW
6934                                        ;*THAT THE MUX SELECTED THE WRONG INPUT TO BE CLOCKED INTO THE PCR.
```

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

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

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

                                        ROMCLK                  ;DO A 'BRANCH ALWAYS' WITH
                                        JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
                                        114400                  ;BAB BITS 11&12 SET THIS SHOULD CLOCK
                                                                ;THESE BITS INTO BITS 8&9 OF THE PCR.
                                        ROMCLK                  ;NOW READ THE PCR HIGH
                                        JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
                                        121265                  ;AND PUT INTO PORTS.
                                                                ;REG. BR NO CLK OF BAB BITS
                                        MOVB    5(R1),R5        ;READ THE PCR.
                                        MOVB    #3,R4           ;EXPECT BITS 8,9 TO BE SET.
                                        BIC     #177774,R5      ;;STRIP ANY JUNK                VRG0182
                                        CMP     R4,R5           ;OK?
                                        BEQ     20$

                                        ERROR   15              ;"BRANCH FIELD H" STUCK HIGH OR
                                        TRAP    C$ERDF
                                        .WORD   15
                                        .WORD   EM0
                                        .WORD   ERR15
                                                                ;OTHER PROBLEM IN THIS AREA.

                                20$:
                                        ENDTST
                                L10112:
                                        TRAP    C$ETST

                                        BADHEAD
                                        ;************** TEST 41 **************
                                        ;*
                                        ;*IN THIS TEST WE'RE GOING TO MAKE SURE THAT ONLY SPO
                                        ;*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.
                                        BADHEAD
                                        ;************** TEST 41 **************

                                        BGNTST
```

```
6991  026544                                T41::
6992  026544                                      MYINT
6993  026544  013701  002516                      MOV     KMCSR,R1            ;RECORD DEVICE ADDR.
6994  026550  005005                              CLR     R5                  ;START WITH ADDR-ZERO
6995
6996  026552  042737  000017  026574  10$:      BIC     #17,20$             ;STRIP SP ADDR FIELD FROM INSTR
6997  026560  010551  000004                      MOV     R5,4(R1)            ;PUT SP ADDR INTO PORT4.
6998  026564  050537  026574                      BIS     R5,20$              ;ADD SP ADDR TO INSTR.
6999  026570                                      ROMCLK
7000  026570  004537  003044                      JSR     R5,..ROMCLK         ;CLOCK INSTRUCTION
7001  026574  123100                      20$:    123100                      ;WRITE TO SP
7002  026576  005205                              INC     R5                  ;UPDATE ADDRESS
7003  026600  120527  000020                      CMPB    R5,#20              ;IF NOT THROUGH, REPEAT.
7004  026604  001362                              BNE     10$
7005
7006  026606                                      ROMCLK                      ;NOW MOV SPO TO OBUS* PORT4
7007  026606  004537  003044                      JSR     R5,..ROMCLK         ;CLOCK INSTRUCTION
7008  026612  061204                              061204                      ;
7009  026614  116104  000004                      MOVB    4(R1),R4            ;READ PORT4 IT S/B ZERO
7010  026620  001410                              BEQ     30$
7011  026622  012705  000000                      MOV     #0,R5
7012  026626                                      ERROR   43                  ;SPO NOT SELECTED FOR SOURCE-SEE
7013  026632  104455                              TRAP    C$ERDF
7014  026634  000053                              .WORD   43
7015  026636  005055                              .WORD   EM0
7016  026640  011006                              .WORD   ERR43
7017                                                                          ;DISCUSSION IN HEADER.
7018
7019  026642                                30$:   ENDTST
7020  026642                                L10113:
7021  026642  104401                              TRAP    C$ETST
7022
7023  026644                                      BADHEAD
7024                                                ;************** TEST 42 **************
7025                                                ;*
7026                                                ;*IN THIS TEST WE ARE GOING TO MAKE SURE THAT THE
7027                                                ;*SIGNAL "MOV INST H" (AND ITS ASSOC. TRIBS) DOESN'T GET
7028                                                ;*STUCK HIGH. IN ORDER TO DO THIS WE'LL CLEAR THE PC HIGH REG
7029                                                ;*PUT KNOWN DATA IN THE BREG AND SP1 THEN WE'LL A BRANCH
7030                                                ;*WITH CROM BITS 0-3 SET AS WELL AS CROM BIT 9 WITH CROM BITS 8 AND 11 CLEAR.
7031                                                ;*IF "MOV INST H" GETS STUCK HIGH, THE PC REG HIGH WILL GET LOADED
7032                                                ;*WITH THE CONTENTS OF THE ALU
7033  026644                                      BADHEAD
7034                                                ;************** TEST 42 **************
7035
7036  026644                                      BGNTST
7037  026644                                T42::
7038  026644                                      SKIP07  10$                 ;ONLY DO IF M8207
7039                                                ;GOTO 10$ IF M8207
7040  026654                                      EXIT TST                    ;ELSE EXIT
7041  026654  104432                              TRAP    C$EXIT
7042  026656  000110                              .WORD   L10114-.
7043
7044  026660                                10$:   MYINT                       ;DO INITIAL TEST SET-UP.
7045  026660  013701  002516                      MOV     KMCSR,R1            ;RECORD DEVICE ADDR.
7046  026664                                      MSTCLR                      ;DO A MASTER CLEAR.
```

```
7047  026670  005737  002470              TST     RUNB
7048  026674  001034                      BNE     20$
7049
7050                                      ;TO RUN THIS SECTION OF CODE YOU MUST TURN SW7 OF SWITCH PACK #E28
7051                                      ;OFF SO THAT M8207 NOT SELFSTARTING.
7052
7053  026676  012761  000002  000004      MOV     #2,4(R1)        ;PUT A 2 INTO SP1
7054  026704                              ROMCLK                  ;PORT4 TO SCRATCH PAD 1
7055  026704  004537  003044              JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
7056  026710  123101                      123101
7057  026712  012761  000004  000004      MOV     #4,4(R1)
7058  026720                              ROMCLK
7059  026720  004537  003044              JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
7060  026724  123100                      123100
7061  026726                              ROMCLK                  ;NOW DO A BRANCH ON C-BIT SET
7062  026726  004537  003044              JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
7063  026732  141201                      141201                  ;BASED ON SP CONTENTS
7064                                                              ;OK-WHAT WE ARE REALLY
7065                                                              ;INTERESTED IN IS SEEING IF THE
7066                                                              ;PC HIGH REG GETS LOADED WITH
7067                                                              ;THE CONTENTS OF THE ALU (2)
7068                                                              ;IF THIS OCCURS, WE CAN PROBABLY
7069                                                              ;SAY THAT "MOV INSTR" REMAINED
7070                                                              ;HIGH.
7071  026734                              ROMCLK                  ;READ PC HIGH, PUT INTO PORT5
7072  026734  004537  003044              JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
7073  026740  121265                      121265
7074  026742  116104  000005              MOVB    5(R1),R4        ;READ PC REG HIGH FROM PORT
7075  026746  001407                      BEQ     20$             ;SHOULD BE CLEAR
7076  026750  005005                      CLR     R5
7077
7078  026752                              ERROR   15              ;ERROR-PC REG HIGH S/B CLEAR-SEE HEADER
7079  026756  104455                      TRAP    C$ERDF
7080  026760  000017                      .WORD   15
7081  026762  005055                      .WORD   EM0
7082  026764  010006                      .WORD   ERR15
7083                                                              ;DISCUSSION.
7084
7085  026766                      20$:
7086  026766                              ENDTST
7087  026766                      L10114:
7088  026766  104401                      TRAP    C$ETST
7089
7090  026770                              BADHEAD
7091                                      ;************** TEST 43 **************
7092                                      ;*TEST THAT MASTER CLEAR, CLEARS BITS IN THE NPR CONTROL REGISTER AND
7093                                      ;*MICROPROCESSOR MISCELLANEOUS REGISTER-FIRST WE'LL SET THE
7094                                      ;*PRIORITY UP SO THAT WHEN WE SET THE BUS REQUEST BIT THAT IT WON'T BUG US
7095                                      ;*THEN WE'LL SET ALL THE BITS IN BOTH REGS EXCEPT THE
7096                                      ;*NPR REQUEST. WE'LL LOOK TO SEE THAT ALL GOT SET, NEXT
7097                                      ;*WE'LL DO A MASTER CLEAR AND BE SURE THAT THEY ALL
7098                                      ;*CLEAR.
7099  026770                              BADHEAD
7100                                      ;************** TEST 43 **************
7101
7102  026770                              BGNTST
```

```
7103  026770                                T43::
7104  026770                                      MYINT
7105  026770  013701  002516                      MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
7106  026774                                      MSTCLR
7107  027000                                      SETPRI  #PRI07          ;DON'T ALLOW INTERRUPTS.
7108  027000  012700  000340                      MOV     #PRI07,R0
7109  027004  104441                             TRAP    C$SPRI
7110  027006  012761  177777  000004              MOV     #-1,4(R1)       ;DATA TO BE SET
7111  027014  042761  000002  000004              BIC     #2,4(R1)        ;DON'T SET AC LOW!
7112  027022                                      ROMCLK
7113  027022  004537  003044                      JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
7114  027026  121111                             121111                  ;PUT INTO MISC REG.
7115  027030  042761  000400  000004              BIC     #400,4(R1)      ;DON'T SET NPR BIT
7116  027036                                      ROMCLK
7117  027036  004537  003044                      JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
7118  027042  121130                             121130                  ;PUT INTO NPR REG
7119  027044                                      ROMCLK
7120  027044  004537  003044                      JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
7121  027050  121225                             121225                  ;MOV MISC REG (11) TO PORT5
7122
7123  027052                                      ROMCLK
7124  027052  004537  003044                      JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
7125  027056  121204                             121204                  ;MOVE NPR REG (10) TO PORT4
7126  027060  012737  146636  002452              MOV     #146636,$GDDAT  ;EXPECT ALL TO SET
7127  027066  016104  000004                      MOV     4(R1),R4        ;READ WHAT HAPPEN
7128  027072  042704  030140                      BIC     #030140,R4      ;MASK UNUSED BITS
7129  027076  023704  002452                      CMP     $GDDAT,R4       ;DID ALL BITS GET SET?
7130  027102  001410                             BEQ     10$             ;YES CONTINUE.
7131  027104                                      BRESET
7132  027104  104433                             TRAP    C$RESET
7133  027106                                      ERROR   46              ;SO SORT OF PROBLEM SETTING BITS
7134  027112  104455                             TRAP    C$ERDF
7135  027114  000056                             .WORD   46
7136  027116  005055                             .WORD   EM0
7137  027120  011214                             .WORD   ERR46
7138                                                                      ;IN THE NPR AND/OR MISC REG.
7139  027122                                      CKLOOP
7140  027122  104406                             TRAP    C$CLP1
7141
7142  027124  152761  000100  000001  10$:       BISB    #100,1(R1)      ;SET MASTER CLEAR
7143  027132  142761  000300  000001              BICB    #300,1(R1)      ;CLEAR MASTER CLEAR
7144
7145  027140                                      ROMCLK
7146  027140  004537  003044                      JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
7147  027144  121225                             121225                  ;MOV MISC REG (11) TO PORT5
7148
7149  027146                                      ROMCLK
7150  027146  004537  003044                      JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
7151  027152  121204                             121204                  ;MOV NPR REG (10) TO PORT4
7152  027154  016104  000004                      MOV     4(R1),R4        ;READ RESULTS
7153  027160  005037  002452                      CLR     $GDDAT          ;EXPECT ZERO
7154  027164  042704  010140                      BIC     #010140,R4      ;MASK UNUSED BITS
7155  027170  001407                             BEQ     20$             ;IF ALL ZERO, EVERYTHING COOL.
7156
7157  027172                                      ERROR   46              ;MASTER CLEAR FAILED TO CLEAR
7158  027176  104455                             TRAP    C$ERDF
```

```
7159  027200  000056                        .WORD   46
7160  027202  005055                        .WORD   EM0
7161  027204  011214                        .WORD   ERR46
7162                                                        ;SOME BITS IN THE NPR AND/OR MISC REGS.
7163  027206                                CKLOOP
7164  027206  104406                        TRAP    C$CLP1
7165
7166  027210                          20$:
7167  027210  012761  000014  000004        MOV     #14,4(R1)       ;NOW WE ARE GOING TO TRY TO
7168  027216                                ROMCLK                  ;SET THE EXT BITS (16&17) IN THE NPR REG.
7169  027216  004537  003044                JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
7170  027222  121110                        121110                  ;IF MASTER CLEAR FAILED TO CLEAR ITSELF
7171  027224                                ROMCLK                  ;THEN WE WILL BE UNABLE TO SET
7172  027224  004537  003044                JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
7173  027230  121205                        121205                  ;THESE BITS
7174  027232  116104  000005                MOVB    5(R1),R4        ;READ REG
7175  027236  042704  000140                BIC     #140,R4         ;MASK UNUSED BITS
7176  027242  012737  000014  002452        MOV     #14,$GDDAT      ;STORE GOOD
7177  027250  023704  002452                CMP     $GDDAT,R4       ;DID BITS SET?
7178  027254  001407                        BEQ     30$             ;YES-CONTINUE
7179
7180  027256                                ERROR   46              ;MASTER CLEAR FAILED TO CLEAR
7181  027262  104455                        TRAP    C$ERDF
7182  027264  000056                        .WORD   46
7183  027266  005055                        .WORD   EM0
7184  027270  011214                        .WORD   ERR46
7185                                                        ;ITSELF, THUS PROHIBITING US FROM
7186                                                        ;FURTHER SETTING BITS IN THE NPR REG.
7187  027272                                CKLOOP
7188  027272  104406                        TRAP    C$CLP1
7189
7190  027274                          30$:  BRESET                  ;NOW WE'LL SEE IF A BUS RESET CLEARS
7191  027274  104433                        TRAP    C$RESET
7192                                                        ;THESE BITS.
7193  027276  005737  002470                TST     RUNB            ;CAN'T DO THIS
7194  027302  001016                        BNE     40$             ;TEST IF RUN SW SET.
7195  027304                                ROMCLK
7196  027304  004537  003044                JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
7197  027310  121204                        121204                  ;READ MISC REG
7198  027312  116104  000004                MOVB    4(R1),R4
7199  027316  001410                        BEQ     40$             ;IF ZERO-END TST
7200
7201  027320  005037  002452                CLR     $GDDAT          ;S/B ZERO
7202
7203  027324                                ERROR   46              ;BUS RESET FAILED TO CLEAR NPR REG
7204  027330  104455                        TRAP    C$ERDF
7205  027332  000056                        .WORD   46
7206  027334  005055                        .WORD   EM0
7207  027336  011214                        .WORD   ERR46
7208                                                        ;MASTER CLEAR WAS ABLE TO LOOK TO THE
7209                                                        ;CIRCUITRY THAT CONVERTS BUS INIT
7210                                                        ;TO "CLEAR"
7211
7212  027340                          40$:
7213  027340                                ENDTST
7214  027340                          L10115:
```

```
  7215  027340  104401                        TRAP    C$ETST
```

```
7216                                    .SBTTL HARDWARE PARAMETER CODING SECTION
7217
7218
7219
7220                                    ;////////////////////////////////////////////////////////////////////////
7221                                    ;/ THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
7222                                    ;/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
7223                                    ;/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
7224                                    ;/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
7225                                    ;/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
7226                                    ;/ WITH THE OPERATOR.
7227                                    ;////////////////////////////////////////////////////////////////////////
7228
7229  027342                                    BGNHRD
7230  027342  000016                            .WORD L10116-L$HARD/2
7231  027344                            L$HARD::
7232
7233  027344                                    GPRMD   WPM,0,0,7,0,7,YES
7234  027344  000032                            .WORD   T$CODE
7235  027346  027400                            .WORD   WPM
7236  027350  000007                            .WORD   7
7237  027352  000000                            .WORD   T$LOLIM
7238  027354  000007                            .WORD   T$HILIM
7239  027356                                    GPRMA   ADDRES,2,0,160000,177776,YES
7240  027356  001031                            .WORD   T$CODE
7241  027360  027452                            .WORD   ADDRES
7242  027362  160000                            .WORD   T$LOLIM
7243  027364  177776                            .WORD   T$HILIM
7244                                    ;       GPRMA   VECTOR,4,0,0,674,YES
7245                                    ;       GPRMD   PRIRTY,6,0,7000,4,7,YES
7246                                    ;       GPRMD   LNUNIT,10,0,3,0,3,YES
7247                                    ;       GPRMD   SWPAC1,12,0,377,0,377,YES
7248                                    ;       GPRMD   SWPAC2,14,0,377,0,377,YES
7249                                    ;       GPRMD   LOOPBK,16,0,40000,0,1,YES
7250  027366                                    GPRMD   ISRUN,24,0,7,0,1,YES
7251  027366  012032                            .WORD   T$CODE
7252  027370  030050                            .WORD   ISRUN
7253  027372  000007                            .WORD   7
7254  027374  000000                            .WORD   T$LOLIM
7255  027376  000001                            .WORD   T$HILIM
7256  027400                                    ENDHRD
7257                                            .EVEN
7258  027400                            L10116:
7259
7260  027400  044127  041511  020110  WPM:      .ASCIZ  "WHICH MICRO-CPU? (0=M8200,4=M8204,7=M8207"
7261  027406  044515  051103  026517
7262  027414  050103  037525  024040
7263  027422  036460  034115  030062
7264  027430  026060  036464  034115
7265  027436  030062  026064  036467
7266  027444  034115  030062  000067
7267  027452  044515  051103  026517  ADDRES: .ASCIZ  /MICRO-CPU  CSR ADDRESS : /
7268  027460  050103  020125  041440
7269  027466  051123  040440  042104
7270  027474  042522  051523  035040
7271  027502  000040
```

```
7272  027504  044515  051103  026517  VECTOR: .ASCIZ  /MICRO-CPU VECTOR ADDRESS : /
7273  027512  050103  020125  042526
7274  027520  052103  051117  040440
7275  027526  042104  042522  051523
7276  027534  035040  000040
7277  027540  044515  051103  026517  PRIRTY: .ASCIZ  /MICRO-CPU PRIORITY LEVEL : /
7278  027546  050103  020125  051120
7279  027554  047511  044522  054524
7280  027562  046040  053105  046105
7281  027570  035040  000040
7282  027574  044127  041511  020110  LNUNIT: .ASCIZ  /WHICH LINE UNIT (0-3)? 0=NONE,1=M8201,2=M8202,3=M8203 : /
7283  027602  044514  042516  052440
7284  027610  044516  020124  030050
7285  027616  031455  037451  030040
7286  027624  047075  047117  026105
7287  027632  036461  034115  030062
7288  027640  026061  036462  034115
7289  027646  030062  026062  036463
7290  027654  034115  030062  020063
7291  027662  020072     000
7292  027665     123  044527  041524  SWPAC1: .ASCIZ  /SWITCH PACK #1 (DDCMP LINE #) : /
7293  027672  020110  040520  045503
7294  027700  021440  020061  042050
7295  027706  041504  050115  046040
7296  027714  047111  020105  024443
7297  027722  035040  000040
7298  027726  053523  052111  044103  SWPAC2: .ASCIZ  /SWITCH PACK #2 (BM873 BOOT ADR) : /
7299  027734  050040  041501  020113
7300  027742  031043  024040  046502
7301  027750  033470  020063  047502
7302  027756  052117  040440  051104
7303  027764  020051  020072     000
7304  027771     127  046111  020114  LOOPBK: .ASCIZ  /WILL TEST CONNECTOR(S) BE USED ? 0=NO,1=YES : /
7305  027776  042524  052123  041440
7306  030004  047117  042516  052103
7307  030012  051117  051450  020051
7308  030020  042502  052440  042523
7309  030026  020104  020077  036460
7310  030034  047516  030454  054475
7311  030042  051505  035040  000040
7312  030050  044515  051103  026517  ISRUN:  .ASCIZ  "MICRO-PROCESSOR RUN SWITCH  TYPE 0 IF OFF, 1 IF ON :"
7313  030056  051120  041517  051505
7314  030064  047523  020122  052522
7315  030072  020116  053523  052111
7316  030100  044103  020040  054524
7317  030106  042520  030040  044440
7318  030114  020106  043117  026106
7319  030122  030440  044440  020106
7320  030130  047117  035040     000
7321
7322        030136                  .EVEN
7323
7324
7325
7326
7327
```

```
7328
```

```
                                .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.
                                ;////////////////////////////////////////////////////////////////////

030136                                  BGNSFT
030136  000000                          .WORD L10117-L$SOFT/2
030140                          L$SOFT::

030140                                  ENDSFT
                                        .EVEN
030140                          L10117:

                                .EVEN


                                                LASTAD
                                                .EVEN
030140  000000                                  .WORD   0
030142  000000                                  .WORD   0
030144                                  L$LAST::

        000001                  .END
```

```
ADDRES  027452           7241    7267#
ADR   = 000020 G         1561#
ADR1    012600           3442#   3471
ADR2    012604           3443#   3468
ADR4    012444           3392#   3419
ADR5    012450           3393#   3416
ASSEMB= 000010           1250
BIT0  = 000001 G         1534#   1881    1882    1895    1896    6044    6046    6047
BIT00 = 000001 G         1523#   1534
BIT01 = 000002 G         1522#   1533
BIT02 = 000004 G         1521#   1532
BIT03 = 000010 G         1520#   1531
BIT04 = 000020 G         1519#   1530
BIT05 = 000040 G         1518#   1529
BIT06 = 000100 G         1517#   1528
BIT07 = 000200 G         1516#   1527
BIT08 = 000400 G         1515#   1526
BIT09 = 001000 G         1514#   1525
BIT1  = 000002 G         1533#   1879    1881    1882    1889    1896    5517    6042    6044    6046
BIT10 = 002000 G         1513#   2013    2034    2042    3396    3447    3499    3528    6175
BIT11 = 004000 G         1512#   6125    6158    6163
BIT12 = 010000 G         1511#   3131
BIT13 = 020000 G         1510#   2016    2037    2045    3142    3399    3450    3502
BIT14 = 040000 G         1509#   6191
BIT15 = 100000 G         1508#   3146
BIT2  = 000004 G         1532#   1882    1896
BIT3  = 000010 G         1531#
BIT4  = 000020 G         1530#
BIT5  = 000040 G         1529#   3927
BIT6  = 000100 G         1528#   1866    1867    3950
BIT7  = 000200 G         1527#   1867
BIT8  = 000400 G         1526#   1989    3925    6125
BIT9  = 001000 G         1525#   6123    6125    6175
BOE   = 000400 G         1565#
CLKX    002360           1617#
CLRALL  003166           1902#   4013    4037    4061    4814    4840    4864    4916    4942    4966    5017    5043    5067
                         5118    5144    5168    5220    5246    5270    5321    5347    5371    5898    5935    5972    6008
CRAMA   003602           2043    2052#
CZDMQ   002000 G         1256#
C$AU  = 000052           1250#   3258
C$AUTO= 000061           1250#   3204
C$BRK = 000022           1250#   3606    3668    3759    3852
C$BSEG= 000004           1250#   3395    3445    3497    3522    4010    4036    4060    4112    4135    4158    4209    4234
                         4259    4311    4335    4359    4412    4436    4460    4513    4537    4561    4613    4637    4661
                         4713    4737    4761    4812    4837    4863    4914    4939    4965    5016    5040    5066    5117
                         5141    5167    5219    5243    5269    5320    5344    5370
C$BSUB= 000002           1250#
C$CEFG= 000045           1250#
C$CLCK= 000062           1250#
C$CLEA= 000012           1250#   3220
C$CLOS= 000035           1250#
C$CLP1= 000006           1250#   7140    7164    7188
C$CVEC= 000036           1250#
C$DCLN= 000044           1250#
C$DODU= 000051           1250#   3198
C$DRPT= 000024           1250#
```

```
C$DU   = 000053  1250#  3240
C$EDIT= 000003  1250#  1327
C$ERDF= 000055  1250#  3297  3345  3365  3404  3455  3511  3534  3594  3656  3718  3751  3816
                3841  3888  3933  3956  3975  4025  4049  4073  4124  4147  4170  4222  4247
                4272  4324  4348  4372  4425  4449  4473  4526  4550  4574  4626  4650  4674
                4726  4750  4774  4826  4852  4876  4928  4954  4978  5029  5055  5079  5130
                5156  5180  5232  5258  5282  5333  5359  5383  5456  5527  5645  5728  5833
                5908  5927  5944  5982  6000  6017  6054  6071  6087  6130  6152  6165  6182
                6195  6333  6364  6386  6412  6484  6539  6629  6644  6659  6673  6689  6701
                6713  6725  6807  6838  6852  6908  6966  7013  7079  7132  7158  7181  7204
C$ERHR= 000056  1250#
C$ERRO= 000060  1250#
C$ERSF= 000054  1250#
C$ERSO= 000057  1250#
C$ESCA= 000010  1250#  3288  3305  3351  3409  3460  3516  3539  3599  3661  3723  3756  3849
                3938  4030  4054  4078  4129  4152  4175  4228  4253  4277  4329  4353  4377
                4430  4454  4478  4531  4555  4579  4631  4655  4679  4731  4755  4779  4831
                4857  4881  4933  4959  4983  5034  5060  5084  5135  5161  5185  5237  5263
                5287  5338  5364  5388  5465  5535  5680  5868  5914  5933  5988  6006  6338
                6391  6417  6546  6635  6651  6665  6679  6695  6706  6718  6843  6915
C$ESEG= 000005  1250#  3413  3464  3520  3543  4034  4058  4082  4133  4156  4179  4232  4257
                4281  4333  4357  4381  4434  4458  4482  4535  4559  4583  4635  4659  4683
                4735  4759  4783  4835  4861  4885  4937  4963  4987  5038  5064  5088  5139
                5165  5189  5241  5267  5291  5342  5368  5392
C$ESUB= 000003  1250#
C$ETST= 000001  1250#  3310  3373  3423  3475  3551  3613  3675  3766  3895  3982  4085  4182
                4284  4384  4485  4586  4686  4786  4888  4990  5091  5192  5294  5395  5473
                5545  5736  5878  5951  6024  6097  6207  6436  6492  6561  6753  6815  6861
                6924  6975  7021  7088  7215
C$EXIT= 000032  1250#  3386  3436  3489  3780  3910  4002  4104  4201  4303  4403  4504  4605
                4705  4805  4907  5009  5110  5212  5313  5412  5493  5588  5776  5893  5967
                6173  6423  6508  6552  6579  6731  6788  6878  6944  7041
C$GETB= 000026  1250#
C$GETW= 000027  1250#
C$GMAN= 000043  1250#
C$GPHR= 000042  1250#  3093
C$GPLO= 000030  1250#
C$GPRI= 000040  1250#
C$INIT= 000011  1250#  3178
C$INLP= 000020  1250#
C$MANI= 000050  1250#
C$MEM  = 000031  1250#
C$MSG  = 000023  1250#  2390  2419  2448  2476  2504  2533  2562  2590  2618  2646  2674  2702
                2731  2759  2782  2810  2838  2850  2869  2886  2903  2923  2942  2967  2986
                3006
C$OPEN= 000034  1250#
C$PNTB= 000014  1250#  2368  2374  2382  2387  2397  2403  2411  2416  2426  2432  2440  2445
                2455  2461  2468  2473  2483  2489  2496  2501  2511  2517  2525  2530  2540
                2546  2554  2559  2569  2575  2582  2587  2597  2603  2610  2615  2625  2631
                2638  2643  2653  2659  2666  2671  2681  2687  2694  2699  2709  2715  2723
                2728  2738  2744  2751  2756  2768  2774  2779  2789  2795  2802  2807  2817
                2823  2830  2835  2846  2865  2882  2899  2919  2938  2957  2963  2982  3002
C$PNTF= 000017  1250#  2859  2876  2893  2913  2932  2951  2976  2996
C$PNTS= 000016  1250#
C$PNTX= 000015  1250#
C$QIO  = 000377  1250#
```

```
C$RDBU= 000007          1250#
C$REFG= 000047          1250#   3058    3064    3070    3076
C$RESE= 000033          1250#   3216    3237    6113    7132    7191
C$REVI= 000003          1250#   1326
C$RFLA= 000021          1250#
C$RPT = 000025          1250#   3027
C$SEFG= 000046          1250#
C$SPRI= 000041          1250#   7109
C$SVEC= 000037          1250#
C$TPRI= 000013          1250#
DFPTBL  002262 G        1447#
DH0     005723          2287#   2770
DH1     005724          2290#   2370    2399    2428    2513    2542
DH2     005755          2295#   2457    2485    2571    2599    2627    2655    2683    2740    2791    2819
DH3     005775          2298#   2711
DIAGMC= 000000          1250
EF.CON= 000036 G        1541#   3069
EF.NEW= 000035 G        1542#   3063
EF.PWR= 000034 G        1543#
EF.RES= 000037 G        1540#   3075
EF.STA= 000040 G        1539#   3057
EMB1    005410          2246#   6089
EMB50   005545          2266#   6056
EMB51   005577          2271#   6073
EM0     005055          2204#   3299    3347    3367    3406    3457    3513    3536    3596    3658    3720    3753    3818
                        3843    3890    3935    3958    3977    4027    4051    4075    4126    4149    4172    4224    4249
                        4274    4326    4350    4374    4427    4451    4475    4528    4552    4576    4628    4652    4676
                        4728    4752    4776    4828    4854    4878    4930    4956    4980    5031    5057    5081    5132
                        5158    5182    5234    5260    5284    5335    5361    5385    5458    5529    5647    5730    5835
                        5910    5929    5946    5984    6002    6019    6132    6154    6167    6184    6197    6335    6366
                        6388    6414    6486    6541    6631    6646    6661    6675    6691    6703    6715    6727    6809
                        6840    6854    6910    6968    7015    7081    7136    7160    7183    7206
EM1     005056          2205#   2364    2422
EM10    005260          2230#   2593    2649
EM11    005312          2235#   2677
EM12    005333          2238#   2705
EM16    005353          2241#   2734
EM17    005434          2251#   2764
EM2     005076          2208#   2393
EM27    005463          2255#
EM29    005512          2260#   2785
EM3     005131          2213#   2451
EM35    005434          2250#   2813
EM4     005144          2215#   2479
EM5     005171          2219#   2507
EM6     005216          2223#   2536
EM7     005233          2226#   2565    2621
ENDBUG  003044          1874#
ENDIT   012040          3072    3164    3168    3175#
ERRFLG  002350          1613#
ERR1    006032 G        2363#   3407    3458    3514    6542
ERR10   007154 G        2564#
ERR11   007276 G        2592#   3819    3844
ERR12   007420 G        2620#   3348    3368
ERR13   007542 G        2648#   5459
ERR14   007664 G        2676#   5530
```

```
ERR15   010006 G        2704#  5648   5836   6969   7082
ERR16   010134 G        2733#  5911   5930   5947   5985   6003   6020
ERR17   010256 G        2763#  6133   6155   6198
ERR2    006160 G        2392#  3537
ERR29   010350 G        2784#  6336   6367   6632   6647   6662   6676   6692   6704   6716   6728
ERR3    006306 G        2421#
ERR35   010472 G        2812#  6168   6185
ERR36   010614 G        2841#  3300
ERR4    006434 G        2450#
ERR40   010636 G        2853#  6810
ERR41   010702 G        2871#  6841
ERR42   010744 G        2888#  6855
ERR43   011006 G        2906#  7016
ERR44   011054 G        2925#  3891
ERR45   011124 G        2944#  5731
ERR46   011214 G        2969#  7137   7161   7184   7207
ERR47   011264 G        2989#  6911
ERR5    006556 G        2478#  4028   4052   4076   4127   4150   4173   4225   4250   4273   4327   4351   4375
                        4428   4452   4476   4529   4553   4577   4629   4653   4677   4729   4753   4777   4829
                        4855   4879   4931   4957   4981   5032   5058   5082   5133   5159   5183   5235   5261
                        5285   5336   5362   5386
ERR6    006700 G        2506#  3597   3659   3721   3754   6415
ERR7    007026 G        2535#  3936   3959   3978   6389   6487
EVL   = 000004 G        1559#
E$END = 002100          1250#
E$LOAD= 000035          1250#  1350
FADR    002412          1630#  2719   5599*  5682*  5787*  5870*
FLAG    002436          1628#  2519   3567*  3571   3572   3607*  3608   3628*  3633   3634   3669*  3670   3691*
                        3692   3695   3696   3725*  3726   3729*  3730   3733   3760*  3761   5592*  5604   5683*
                        5780*  5792   5871*
FM1     005046          2199#  2365   2371   2394   2400   2423   2429   2452   2458   2480   2486   2508   2514
                        2537   2543   2566   2572   2594   2600   2622   2628   2650   2656   2678   2684   2706
                        2712   2735   2741   2765   2771   2786   2792   2814   2820
FTIME   002462          1650#  3047   3051*
F$AU  = 000015          1250#  3255   3257
F$AUTO= 000020          1250#  3182   3203
F$BGN = 000040          1250#  1256   1370   2363   2392   2421   2450   2478   2506   2535   2564   2592   2620
                        2648   2676   2704   2733   2763   2784   2812   2841   2853   2871   2888   2906   2925
                        2944   2969   2989   3017   3023   3041   3182   3214   3234   3255   3280   3288   3305
                        3309   3323   3351   3372   3383   3386   3395   3409   3422   3433   3436   3445   3460
                        3474   3486   3489   3497   3516   3522   3539   3550   3562   3599   3612   3623   3661
                        3674   3686   3723   3756   3765   3777   3780   3849   3894   3907   3910   3938   3981
                        3998   4002   4010   4030   4036   4054   4060   4078   4084   4101   4104   4112   4129
                        4135   4152   4158   4175   4181   4198   4201   4209   4228   4234   4253   4259   4277
                        4283   4300   4303   4311   4329   4335   4353   4359   4377   4383   4400   4403   4412
                        4430   4436   4454   4460   4478   4484   4501   4504   4513   4531   4537   4555   4561
                        4579   4585   4602   4605   4613   4631   4637   4655   4661   4679   4685   4702   4705
                        4713   4731   4737   4755   4761   4779   4785   4802   4805   4812   4831   4837   4857
                        4863   4881   4887   4904   4907   4914   4933   4939   4959   4965   4983   4989   5006
                        5009   5016   5034   5040   5060   5066   5084   5090   5107   5110   5117   5135   5141
                        5161   5167   5185   5191   5209   5212   5219   5237   5243   5263   5269   5287   5293
                        5310   5313   5320   5338   5344   5364   5370   5388   5394   5409   5412   5465   5472
                        5490   5493   5535   5544   5583   5588   5620   5735   5771   5776   5868   5877   5890
                        5893   5914   5933   5950   5964   5967   5988   6006   6023   6035   6096   6111   6173
                        6206   6221   6338   6391   6417   6423   6435   6449   6491   6504   6508   6546   6552
                        6560   6574   6579   6635   6651   6665   6679   6695   6706   6718   6731   6752   6767
```

```
                    6788    6814    6828    6843    6860    6874    6878    6915    6923    6940    6944    6974    6991
                    7020    7037    7041    7087    7103    7214    7230    7342
F$CLEA= 000007      1250#   3214    3219
F$DU  = 000016      1250#   3234    3239
F$END = 000041      1250#   1256    2391    2420    2449    2477    2505    2534    2563    2591    2619    2647    2675
                    2703    2732    2760    2783    2811    2839    2851    2870    2887    2904    2924    2943    2968
                    2987    3007    3021    3028    3179    3205    3221    3241    3259    3280    3288    3305    3309
                    3311    3323    3351    3372    3374    3383    3386    3409    3414    3422    3424    3433    3436
                    3460    3465    3474    3476    3486    3489    3516    3521    3539    3544    3550    3552    3562
                    3599    3612    3614    3623    3661    3674    3676    3686    3723    3756    3765    3767    3777
                    3780    3849    3894    3896    3907    3910    3938    3981    3983    3998    4002    4030    4035
                    4054    4059    4078    4083    4084    4086    4101    4104    4129    4134    4152    4157    4175
                    4180    4181    4183    4198    4201    4228    4233    4253    4258    4277    4282    4283    4285
                    4300    4303    4329    4334    4353    4358    4377    4382    4383    4385    4400    4403    4430
                    4435    4454    4459    4478    4483    4484    4486    4501    4504    4531    4536    4555    4560
                    4579    4584    4585    4587    4602    4605    4631    4636    4655    4660    4679    4684    4685
                    4687    4702    4705    4731    4736    4755    4760    4779    4784    4785    4787    4802    4805
                    4831    4836    4857    4862    4881    4886    4887    4889    4904    4907    4933    4938    4959
                    4964    4983    4988    4989    4991    5006    5009    5034    5039    5060    5065    5084    5089
                    5090    5092    5107    5110    5135    5140    5161    5166    5185    5190    5191    5193    5209
                    5212    5237    5242    5263    5268    5287    5292    5293    5295    5310    5313    5338    5343
                    5364    5369    5388    5393    5394    5396    5409    5412    5465    5472    5474    5490    5493
                    5535    5544    5546    5583    5588    5680    5735    5737    5771    5776    5868    5877    5879
                    5890    5893    5914    5933    5950    5952    5964    5967    5988    6006    6023    6025    6035
                    6096    6098    6111    6173    6206    6208    6221    6338    6391    6417    6423    6435    6437
                    6449    6491    6493    6504    6508    6546    6552    6560    6562    6574    6579    6635    6651
                    6665    6679    6695    6706    6718    6731    6752    6754    6767    6788    6814    6816    6828
                    6843    6860    6862    6874    6878    6915    6923    6925    6940    6944    6974    6976    6991
                    7020    7022    7037    7041    7087    7089    7103    7214    7216    7259    7349
F$HARD= 000004      1250#   7230    7257
F$HW  = 000013      1250#   1445    1463
F$INIT= 000006      1250#   3041    3177
F$JMP = 000050      1250#   3021    3386    3436    3489    3780    3910    4002    4104    4201    4303    4403    4504
                    4605    4705    4805    4907    5009    5110    5212    5313    5412    5493    5588    5776    5893
                    5967    6173    6423    6508    6552    6579    6731    6788    6878    6944    7041
F$MOD = 000000      1250#   1256
F$MSG = 000011      1250#   2363    2389    2392    2418    2421    2447    2450    2475    2478    2503    2506    2532
                    2535    2561    2564    2589    2592    2617    2620    2645    2648    2673    2676    2701    2704
                    2730    2733    2758    2763    2781    2784    2809    2812    2837    2841    2849    2853    2868
                    2871    2885    2888    2902    2906    2922    2925    2941    2944    2966    2969    2985    2989
                    3005
F$PROT= 000021      1250#   1370    1375
F$PWR = 000017      1250#
F$RPT = 000012      1250#   3017    3026
F$SEG = 000003      1250#   3395    3412    3445    3463    3497    3519    3522    3542    4010    4033    4036    4057
                    4060    4081    4112    4132    4135    4155    4158    4178    4209    4231    4234    4256    4259
                    4280    4311    4332    4335    4356    4359    4380    4412    4433    4436    4457    4460    4481
                    4513    4534    4537    4558    4561    4582    4613    4634    4637    4658    4661    4682    4713
                    4734    4737    4758    4761    4782    4812    4834    4837    4860    4863    4884    4914    4936
                    4939    4962    4965    4986    5016    5037    5040    5063    5066    5087    5117    5138    5141
                    5164    5167    5188    5219    5240    5243    5266    5269    5290    5320    5341    5344    5367
                    5370    5391
F$SOFT= 000005      1250#   7342    7347
F$SRV = 000010      1250#
F$SUB = 000002      1250#
F$SW  = 000014      1250#   1472    1478
```

```
F$TEST= 000001          1250#  3281   3309   3324   3372   3384   3422   3434   3474   3487   3550   3563   3612
                        3624   3674   3687   3765   3778   3894   3908   3981   3999   4084   4102   4181   4199
                        4283   4301   4383   4401   4484   4502   4585   4603   4685   4703   4785   4803   4887
                        4905   4989   5007   5090   5108   5191   5210   5293   5311   5394   5410   5472   5491
                        5544   5584   5735   5772   5877   5891   5950   5965   6023   6036   6096   6112   6206
                        6222   6435   6450   6491   6505   6560   6575   6752   6768   6814   6829   6860   6875
                        6923   6941   6974   6992   7020   7038   7087   7104   7214
GETPRM  011462          3078   3087#  3096
G$CNTO= 000200          1250#
G$DELM= 000372          1250#
G$DISP= 000003          1250#
G$EXCP= 000400          1250#
G$HILI= 000002          1250#
G$LOLI= 000001          1250#
G$NO  = 000000          1250#
G$OFFS= 000400          1250#  7234   7240   7251
G$OFSI= 000376          1250#  7234   7240   7251
G$PRMA= 000001          1250#  7240
G$PRMD= 000002          1250#  7234   7251
G$PRML= 000000          1250#
G$RADA= 000140          1250#
G$RADB= 000000          1250#
G$RADD= 000040          1250#
G$RADL= 000120          1250#
G$RADO= 000020          1250#  7234   7240   7251
G$XFER= 000004          1250#
G$YES = 000010          1250#  7234   7240   7251
HELP  = 000000          1241#  1280   1369   1430   1739   3019   3024
HOL   = 100000 G        1572#
IBE   = 010000 G        1569#
IDU   = 000040 G        1562#
IER   = 020000 G        1570#
INIFLG  002474          1659#
INSTU   003616          2044   2055#
ISR   = 000100 G        1563#
ISRUN   030050          7252   7312#
IXE   = 004000 G        1568#
I$AU  = 000041          1250#  3255#  3259#
I$AUTO= 000041          1250#  3182#  3205#
I$CLN = 000041          1250#  3214#  3221#
I$DU  = 000041          1250#  3234#  3241#
I$HRD = 000041          7230#  7259#
I$INIT= 000041          1250#  3041#  3179#
I$MOD = 000040          1250#  1256#
I$MSG = 000041          1250#  2363#  2391#  2392#  2420#  2421#  2449#  2450#  2477#  2478#  2505#  2506#  2534#
                        2535#  2563#  2564#  2591#  2592#  2619#  2620#  2647#  2648#  2675#  2676#  2703#  2704#
                        2732#  2733#  2760#  2763#  2783#  2784#  2811#  2812#  2839#  2841#  2851#  2853#  2870#
                        2871#  2887#  2888#  2904#  2906#  2924#  2925#  2943#  2944#  2968#  2969#  2987#  2989#
                        3007#
I$PROT= 000040          1250#  1370#
I$PTAB= 000041          1250#
I$PWR = 000041          1250#
I$RPT = 000041          1250#  3017#  3028#
I$SEG = 000041          1250#  3280   3323   3383   3395#  3409   3414#  3433   3445#  3460   3465#  3486   3497#
                        3516   3521#  3522#  3539   3544#  3562   3623   3686   3777   3907   3998   4010#  4030
                        4035#  4036#  4054   4059#  4060#  4078   4083#  4101   4112#  4129   4134#  4135#  4152
```

```
                    4157#  4158#  4175   4180#  4198   4209#  4228   4233#  4234#  4253   4258#  4259#  4277
                    4282#  4300   4311#  4329   4334#  4335#  4353   4358#  4359#  4377   4382#  4400   4412#
                    4430   4435#  4436#  4454   4459#  4460#  4478   4483#  4501   4513#  4531   4536#  4537#
                    4555   4560#  4561#  4579   4584#  4602   4613#  4631   4636#  4637#  4655   4660#  4661#
                    4679   4684#  4702   4713#  4731   4736#  4737#  4755   4760#  4761#  4779   4784#  4802
                    4812#  4831   4836#  4837#  4857   4862#  4863#  4881   4886#  4904   4914#  4933   4938#
                    4939#  4959   4964#  4965#  4983   4988#  5006   5016#  5034   5039#  5040#  5060   5065#
                    5066#  5084   5089#  5107   5117#  5135   5140#  5141#  5161   5166#  5167#  5185   5190#
                    5209   5219#  5237   5242#  5243#  5263   5268#  5269#  5287   5292#  5310   5320#  5338
                    5343#  5344#  5364   5369#  5370#  5388   5393#  5409   5490   5587   5771   5890   5964
                    6035   6111   6221   6449   6504   6574   6767   6828   6874   6940   6991   7037   7103
I$SETU= 000041       1250#
I$SFT = 000041       7342#  7349#
I$SRV = 000041       1250#
I$SUB = 000041       1250#  3280   3323   3383   3433   3486   3562   3623   3686   3777   3907   3998   4101
                    4198   4300   4400   4501   4602   4702   4802   4904   5006   5107   5209   5310   5409
                    5490   5583   5771   5890   5964   6035   6111   6221   6449   6504   6574   6767   6828
                    6874   6940   6991   7037   7103

I$TST = 000041       1250#  3280#  3288   3305   3309#  3311#  3323#  3351   3372#  3374#  3383#  3386   3422#
                    3424#  3433#  3436   3474#  3476#  3486#  3489   3550#  3552#  3562#  3599   3612#  3614#
                    3623#  3661   3674#  3676#  3686#  3723   3756   3765#  3767#  3777#  3780   3849   3894#
                    3896#  3907#  3910   3938   3981#  3983#  3998#  4002   4084#  4086#  4101#  4104   4181#
                    4183#  4198#  4201   4283#  4285#  4300#  4303   4383#  4385#  4400#  4403   4484#  4486#
                    4501#  4504   4585#  4587#  4602#  4605   4685#  4687#  4702#  4705   4785#  4787#  4802#
                    4805   4887#  4889#  4904#  4907   4989#  4991#  5006#  5009   5090#  5092#  5107#  5110
                    5191#  5193#  5209#  5212   5293#  5295#  5310#  5313   5394#  5396#  5409#  5412   5465
                    5472#  5474#  5490#  5493   5535   5544#  5546#  5583#  5588   5680   5735#  5737#  5771#
                    5776   5868   5877#  5879#  5890#  5893   5914   5933   5950#  5952#  5964#  5967   5988
                    6006   6023#  6025#  6035#  6096#  6098#  6111#  6173   6206#  6208#  6221#  6338   6391
                    6417   6423   6435#  6437#  6449#  6491#  6493#  6504#  6508   6546   6552   6560#  6562#
                    6574#  6579   6635   6651   6665   6679   6695   6706   6718   6731   6752#  6754#  6767#
                    6788   6814#  6816#  6828#  6843   6860#  6862#  6874#  6878   6915   6923#  6925#  6940#
                    6944   6974#  6976#  6991#  7020#  7022#  7037#  7041   7087#  7089#  7103#  7214#  7216#
J$JMP = 000167       1250#  3021
KMACTV  002376       1624#
KMCSR   002516       1695#  3099*  3184   3281   3326   3389   3439   3492   3564   3625   3688   3783   3913
                    4006   4107   4204   4306   4406   4507   4608   4708   4808   4910   5012   5113   5215
                    5316   5415   5496   5585   5773   5897   5971   6037   6115   6147   6223   6451   6512
                    6583   6769   6830   6882   6948   6993   7045   7105
KMCSRH  002520       1696#  1866*  1867*  1879*  1881*  1882*  1889*  1893*  1895*  1896*  1899*  3101*  3102*
KMCTL   002522       1697#  3104*  3105*
KMNUM   002400       1625#
KMPO4   002524       1698#  3107*  3108*
KMPO6   002526       1699#  1880*  1890*  3110*  3111*
KMRLVL  002510       1692#  3115*  3116*
KMRVEC  002506       1691#  3113*
KMTLVL  002514       1694#  3121*  3122*
KMTVEC  002512       1693#  3118*  3119*
LNUNIT  027574       7282#
LOADER  026006       6585   6739#
LOCK    002340       1605#  3728*
LOE   = 040000 G     1571#
LOGDEV  002342       1610#  3082*  3088*  3089   3092   3197
LOKFLG  002476       1661#
LOOPBK  027771       7304#
LOT   = 000010 G     1560#
```

```
L$ACP    002110 G          1357#
L$APT    002036 G          1315#
L$AU     012144 G          1342    3255#
L$AUT    002070 G          1341#
L$AUTO   012042 G          1358    3182#
L$CCP    002106 G          1355#
L$CLEA   012134 G          1356    3214#
L$CO     002032 G          1311#
L$DEPO   002011 G          1293#
L$DESC   002312 G          1348    1594#
L$DESP   002076 G          1347#
L$DEVP   002060 G          1333#
L$DISP   002132 G          1318    1385#
L$DLY    002116 G          1363#
L$DTP    002040 G          1317#
L$DTYP   002034 G          1313#
L$DU     012140 G          1344    3234#
L$DUT    002072 G          1343#
L$DVTY   002730 G          1334    1727#
L$EF     002052 G          1328#
L$ENVI   002044 G          1321#
L$ETP    002102 G          1351#
L$EXP1   002046 G          1323#
L$EXP4   002064 G          1337#
L$EXP5   002066 G          1339#
L$HARD   027344 G          1300    7230    7231#
L$HIME   002120 G          1365#
L$HPCP   002016 G          1299#
L$HPTP   002022 G          1303#
L$HW     002262 G          1304    1445    1446#
L$ICP    002104 G          1353#
L$INIT   011340 G          1354    3041#
L$LADP   002026 G          1307#
L$LAST   030144 G          1308    7361#
L$LOAD   002100 G          1349#
L$LUN    002074 G          1345#
L$MREV   002050 G          1325#
L$NAME   002000 G          1282#
L$PRIO   002042 G          1319#
L$PROT   002122 G          1360    1370#
L$PRT    002112 G          1359#
L$REPP   002062 G          1335#
L$REV    002010 G          1291#
L$RPT    011332 G          3017#
L$SOFT   030140 G          7342    7343#
L$SPC    002056 G          1331#
L$SPCP   002020 G          1301#
L$SPTP   002024 G          1305#
L$STA    002030 G          1309#
L$SW     002312 G          1472    1473#
L$TEST   002114 G          1361#
L$TIML   002014 G          1297#
L$UNIT   002012 G          1295#   3089
L10001   002310            1445    1463#
L10002   002312            1472    1478#
L10003   006156            2389#
```

```
L10004  006304            2418#
L10005  006432            2447#
L10006  006554            2475#
L10007  006676            2503#
L10010  007024            2532#
L10011  007152            2561#
L10012  007274            2589#
L10013  007416            2617#
L10014  007540            2645#
L10015  007662            2673#
L10016  010004            2701#
L10017  010132            2730#
L10020  010254            2758#
L10021  010346            2781#
L10022  010470            2809#
L10023  010612            2837#
L10024  010634            2849#
L10025  010700            2868#
L10026  010742            2885#
L10027  011004            2902#
L10030  011052            2922#
L10031  011122            2941#
L10032  011212            2966#
L10033  011262            2985#
L10034  011330            3005#
L10035  011336            3022    3026#
L10036  012040            3177#
L10037  012132            3203#
L10040  012136            3219#
L10041  012142            3239#
L10042  012144            3257#
L10043  012254            3289    3304    3309#
L10044  012420            3352    3372#
L10045  012550            3387    3422#
L10046  012710            3437    3474#
L10047  013152            3490    3550#
L10050  013354            3600    3612#
L10051  013566            3662    3674#
L10052  014110            3724    3757    3765#
L10053  014466            3781    3850    3894#
L10054  014734            3911    3939    3981#
L10055  015200            4003    4084#
L10056  015430            4105    4181#
L10057  015674            4202    4283#
L10060  016140            4304    4383#
L10061  016404            4404    4484#
L10062  016650            4505    4585#
L10063  017114            4606    4685#
L10064  017360            4706    4785#
L10065  017640            4806    4887#
L10066  020120            4908    4989#
L10067  020374            5010    5090#
L10070  020650            5111    5191#
L10071  021126            5213    5293#
L10072  021402            5314    5394#
L10073  021574            5413    5466    5472#
```

```
L10074  021742          5494    5536    5544#
L10075  022320          5589    5681    5735#
L10076  022556          5777    5869    5877#
L10077  022772          5894    5915    5934    5950#
L10100  023206          5968    5989    6007    6023#
L10101  023450          6096#
L10102  024042          6174    6206#
L10103  025114          6339    6392    6418    6424    6435#
L10104  025214          6491#
L10105  025362          6509    6547    6553    6560#
L10106  026040          6580    6636    6652    6666    6680    6696    6707    6719    6732    6752#
L10107  026200          6789    6814#
L10110  026306          6844    6860#
L10111  026444          6879    6916    6923#
L10112  026542          6945    6974#
L10113  026642          7020#
L10114  026766          7042    7087#
L10115  027340          7214#
L10116  027400          7230    7258#
L10117  030140          7342    7348#
MASKX   002362          1618#
MEMLIM  002374          1623#
MEMSET  003474          2023#   4011    4110    4207    4309    4409    4510    4610    4711    4810    4912    5014    5115
                        5217    5318
MEMSZ   002436          1640#   2018    2039    3161*   3169*   3418    3470    3524    3546    3608    3670    3726    3761
                        3780    3787    3834    5412    5493    5588    5776    5893    5967    6522    6548
MRO     002434          1639#   2070    3391*   3397    3417*   3418    3441*   3448    3469*   3470    3495*   3498    3500
                        3501    3508    3523*   3524    3526*   3527    3529    3531    3545*   3546    3568*   3579    3589
                        3603*   3629*   3630*   3641    3651    3663*   3665*   6329*
NEWST   011454          3060    3066    3080#   3090
NEXT    002336          1604#
ONE     002372          1622#
O$APTS= 000000          1250#   1309
O$AU  = 000001          1250#   1279#   1341
O$BGNR= 000000          1250#   1335
O$BGNS= 000000          1250#   1301
O$DU  = 000001          1250#   1279#   1343
O$ERRT= 000000          1250#   1351
O$GNSW= 000000          1250#   1305
O$POIN= 000001          1250#   1279#   1367
O$SETU= 000000          1250#   1295    7359
PATCH   002774          1870#
PNT   = 001000 G        1566#
PRI   = 002000 G        1567#
PRIRTY  027540          7277#
PR100 = 000000 G        1555#
PR101 = 000040 G        1554#
PR102 = 000100 G        1553#
PR103 = 000140 G        1552#
PR104 = 000200 G        1551#
PR105 = 000240 G        1550#
PR106 = 000300 G        1549#
PR107 = 000340 G        1548#   7108
PSTACK  002344          1611#   3046*   6157
QV.FLG  002477          1662#
RAMDAT  003330          1977#   4020    4044    4068    4119    4142    4165    4217    4242    4267    4319    4343    4367
```

```
                    4420    4444    4468    4521    4545    4569    4621    4645    4669    4721    4745    4769    4821
                    4847    4871    4923    4949    4973    5024    5050    5074    5125    5151    5175    5227    5253
                    5277    5328    5354    5378
REGADR  002530      1703#
RETADR  002352      1614#
ROMMAP  012146      2012    3270#   6514    6528
RUN     002410      1629#
RUNB    002470      1653#   3173*   3862    5693    6160    7047    7193
RUNINH  002472      1654#   3124*   3173    3864    5695
SAVACT  002402      1626#
SAVE4   002464      1651#   3049*   3052    3200    3302    6785
SAVE6   002466      1652#   3050*   3053    3201    3303    6786
SAVNUM  002404      1627#
SAVPC   002366      1620#
SAVSP   002364      1619#
SETBR0  003220      1918#   4413    4437    4461
SETBR1  003230      1925#   4514    4538    4562
SETBR4  003240      1933#   4614    4638    4662
SETBR7  003250      1941#   4714    4738    4762
SETC    003260      1949#   4210    4235    4260    4813    5990
SETZ    003312      1966#   4312    4336    4360    4915    5916
SFPTBL  002312 G    1474#
SSTACK  002730      1707#   3044
STAT    002356      1616#
STAT1   002500      1684#   3125*   3127*   3131*   3142*   3146*   3149*
STAT2   002502      1685#   3152*   3154*
STAT3   002504      1686#
STM     005627      2276#   2384    2413    2442    2470    2498    2527    2556    2584    2612    2640    2668    2696
                    2725    2753    2776    2804    2832    2843    2862    2879    2896    2916    2935    2960    2979
                    2999
STRTSW  002354      1615#
SUBRPC  002346      1612#
SVCGBL= 000000      1250#   1256    1263#   1282    1283    1291    1292    1293    1294    1295    1296    1297    1298
                    1299    1300    1301    1302    1303    1304    1305    1306    1307    1308    1309    1310    1311
                    1312    1313    1314    1315    1316    1317    1318    1319    1320    1321    1322    1323    1324
                    1325    1326    1328    1329    1331    1332    1333    1334    1335    1336    1337    1338    1339
                    1340    1341    1342    1343    1344    1345    1346    1347    1348    1349    1350    1351    1352
                    1353    1354    1355    1356    1357    1358    1359    1360    1361    1362    1363    1364    1365
                    1366    1370    1371    1385    1386    1446    1447    1448    1473    1474    1475    1594    1595
                    1727    1728    2363    2364    2392    2393    2421    2422    2450    2451    2478    2479    2506
                    2507    2535    2536    2564    2565    2592    2593    2620    2621    2648    2649    2676    2677
                    2704    2705    2733    2734    2763    2764    2784    2785    2812    2813    2841    2842    2853
                    2854    2871    2872    2888    2889    2906    2907    2925    2926    2944    2945    2969    2970
                    2989    2990    3017    3018    3041    3042    3182    3183    3214    3215    3234    3235    3255
                    3256    7231    7232    7343    7344    7361#   7362
SVCINS= 000000      1250#   1260#   1283    1284    1285    1286    1287    1288    1289    1290    1291    1292    1293
                    1294    1295    1296    1297    1298    1299    1300    1301    1302    1303    1304    1305    1306
                    1307    1308    1309    1310    1311    1312    1313    1314    1315    1316    1317    1318    1319
                    1320    1321    1322    1323    1324    1325    1326    1327    1328    1329    1330    1331    1332
                    1333    1334    1335    1336    1337    1338    1339    1340    1341    1342    1343    1344    1345
                    1346    1347    1348    1349    1350    1351    1352    1353    1354    1355    1356    1357    1358
                    1359    1360    1361    1362    1363    1364    1365    1366    1367    1384    1385    1386    1387
                    1388    1389    1390    1391    1392    1393    1394    1395    1396    1397    1398    1399    1400
                    1401    1402    1403    1404    1405    1406    1407    1408    1409    1410    1411    1412    1413
                    1414    1415    1416    1417    1418    1419    1420    1421    1422    1423    1424    1425    1426
                    1427    1428    1429    1445    1446    1472    1473    1595    1599    1600    1728    1732    1733
```

2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376
2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389
2390	2391	2393	2394	2395	2396	2397	2398	2399	2400	2401	2402	2403
2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	2416
2417	2418	2419	2420	2422	2423	2424	2425	2426	2427	2428	2429	2430
2431	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443
2444	2445	2446	2447	2448	2449	2451	2452	2453	2454	2455	2456	2457
2458	2459	2460	2461	2462	2463	2464	2465	2466	2467	2468	2469	2470
2471	2472	2473	2474	2475	2476	2477	2479	2480	2481	2482	2483	2484
2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495	2496	2497
2498	2499	2500	2501	2502	2503	2504	2505	2507	2508	2509	2510	2511
2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524
2525	2526	2527	2528	2529	2530	2531	2532	2533	2534	2536	2537	2538
2539	2540	2541	2542	2543	2544	2545	2546	2547	2548	2549	2550	2551
2552	2553	2554	2555	2556	2557	2558	2559	2560	2561	2562	2563	2565
2566	2567	2568	2569	2570	2571	2572	2573	2574	2575	2576	2577	2578
2579	2580	2581	2582	2583	2584	2585	2586	2587	2588	2589	2590	2591
2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604	2605
2606	2607	2608	2609	2610	2611	2612	2613	2614	2615	2616	2617	2618
2619	2621	2622	2623	2624	2625	2626	2627	2628	2629	2630	2631	2632
2633	2634	2635	2636	2637	2638	2639	2640	2641	2642	2643	2644	2645
2646	2647	2649	2650	2651	2652	2653	2654	2655	2656	2657	2658	2659
2660	2661	2662	2663	2664	2665	2666	2667	2668	2669	2670	2671	2672
2673	2674	2675	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686
2687	2688	2689	2690	2691	2692	2693	2694	2695	2696	2697	2698	2699
2700	2701	2702	2703	2705	2706	2707	2708	2709	2710	2711	2712	2713
2714	2715	2716	2717	2718	2719	2720	2721	2722	2723	2724	2725	2726
2727	2728	2729	2730	2731	2732	2734	2735	2736	2737	2738	2739	2740
2741	2742	2743	2744	2745	2746	2747	2748	2749	2750	2751	2752	2753
2754	2755	2756	2757	2758	2759	2760	2764	2765	2766	2767	2768	2769
2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	2780	2781	2782
2783	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795	2796
2797	2798	2799	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809
2810	2811	2813	2814	2815	2816	2817	2818	2819	2820	2821	2822	2823
2824	2825	2826	2827	2828	2829	2830	2831	2832	2833	2834	2835	2836
2837	2838	2839	2843	2844	2845	2846	2847	2848	2850	2851	2855	2856
2857	2858	2859	2860	2861	2862	2863	2864	2865	2866	2867	2869	2870
2873	2874	2875	2876	2877	2878	2879	2880	2881	2882	2883	2884	2886
2887	2890	2891	2892	2893	2894	2895	2896	2897	2898	2899	2900	2901
2903	2904	2908	2909	2910	2911	2912	2913	2914	2915	2916	2917	2918
2919	2920	2921	2923	2924	2927	2928	2929	2930	2931	2932	2933	2934
2935	2936	2937	2938	2939	2940	2942	2943	2946	2947	2948	2949	2950
2951	2952	2953	2954	2955	2956	2957	2958	2959	2960	2961	2962	2963
2964	2965	2967	2968	2971	2972	2973	2974	2975	2976	2977	2978	2979
2980	2981	2982	2983	2984	2986	2987	2991	2992	2993	2994	2995	2996
2997	2998	2999	3000	3001	3002	3003	3004	3006	3007	3021	3022	3023
3027	3028	3057	3058	3059	3060	3061	3063	3064	3065	3066	3067	3069
3070	3071	3072	3073	3075	3076	3077	3078	3079	3092	3093	3094	3095
3096	3097	3178	3179	3197	3198	3199	3204	3205	3216	3217	3220	3221
3237	3238	3240	3241	3258	3259	3288	3289	3290	3297	3298	3299	3300
3301	3305	3306	3307	3310	3311	3345	3346	3347	3348	3349	3351	3352
3353	3365	3366	3367	3368	3369	3373	3374	3386	3387	3388	3395	3396
3404	3405	3406	3407	3408	3409	3410	3411	3413	3414	3423	3424	3436
3437	3438	3445	3446	3455	3456	3457	3458	3459	3460	3461	3462	3464
3465	3475	3476	3489	3490	3491	3497	3498	3511	3512	3513	3514	3515
3516	3517	3518	3520	3521	3522	3523	3534	3535	3536	3537	3538	3539

3540	3541	3543	3544	3551	3552	3594	3595	3596	3597	3598	3599	3600
3601	3606	3607	3613	3614	3656	3657	3658	3659	3660	3661	3662	3663
3668	3669	3675	3676	3718	3719	3720	3721	3722	3723	3724	3725	3751
3752	3753	3754	3755	3756	3757	3758	3759	3760	3766	3767	3780	3781
3782	3816	3817	3818	3819	3820	3841	3842	3843	3844	3845	3849	3850
3851	3852	3853	3888	3889	3890	3891	3892	3895	3896	3910	3911	3912
3933	3934	3935	3936	3937	3938	3939	3940	3956	3957	3958	3959	3960
3975	3976	3977	3978	3979	3982	3983	4002	4003	4004	4010	4011	4025
4026	4027	4028	4029	4030	4031	4032	4034	4035	4036	4037	4049	4050
4051	4052	4053	4054	4055	4056	4058	4059	4060	4061	4073	4074	4075
4076	4077	4078	4079	4080	4082	4083	4085	4086	4104	4105	4106	4112
4113	4124	4125	4126	4127	4128	4129	4130	4131	4133	4134	4135	4136
4147	4148	4149	4150	4151	4152	4153	4154	4156	4157	4158	4159	4170
4171	4172	4173	4174	4175	4176	4177	4179	4130	4182	4183	4201	4202
4203	4209	4210	4222	4223	4224	4225	4226	4228	4229	4230	4232	4233
4234	4235	4247	4248	4249	4250	4251	4253	4254	4255	4257	4258	4259
4260	4272	4273	4274	4275	4276	4277	4278	4279	4281	4282	4284	4285
4303	4304	4305	4311	4312	4324	4325	4326	4327	4328	4329	4330	4331
4333	4334	4335	4336	4348	4349	4350	4351	4352	4353	4354	4355	4357
4358	4359	4360	4372	4373	4374	4375	4376	4377	4378	4379	4381	4382
4384	4385	4403	4404	4405	4412	4413	4425	4426	4427	4428	4429	4430
4431	4432	4434	4435	4436	4437	4449	4450	4451	4452	4453	4454	4455
4456	4458	4459	4460	4461	4473	4474	4475	4476	4477	4478	4479	4480
4482	4483	4485	4486	4504	4505	4506	4513	4514	4526	4527	4528	4529
4530	4531	4532	4533	4535	4536	4537	4538	4550	4551	4552	4553	4554
4555	4556	4557	4559	4560	4561	4562	4574	4575	4576	4577	4578	4579
4580	4581	4583	4584	4586	4587	4605	4606	4607	4613	4614	4626	4627
4628	4629	4630	4631	4632	4633	4635	4636	4637	4638	4650	4651	4652
4653	4654	4655	4656	4657	4659	4660	4661	4662	4674	4675	4676	4677
4678	4679	4680	4681	4683	4684	4686	4687	4705	4706	4707	4713	4714
4726	4727	4728	4729	4730	4731	4732	4733	4735	4736	4737	4738	4750
4751	4752	4753	4754	4755	4756	4757	4759	4760	4761	4762	4774	4775
4776	4777	4778	4779	4780	4781	4783	4784	4786	4787	4805	4806	4807
4812	4813	4826	4827	4828	4829	4830	4831	4832	4833	4835	4836	4837
4838	4852	4853	4854	4855	4856	4857	4858	4859	4861	4862	4863	4864
4876	4877	4878	4879	4880	4881	4882	4883	4885	4886	4888	4889	4907
4908	4909	4914	4915	4928	4929	4930	4931	4932	4933	4934	4935	4937
4938	4939	4940	4954	4955	4956	4957	4958	4959	4960	4961	4963	4964
4965	4966	4978	4979	4980	4981	4982	4983	4984	4985	4987	4988	4990
4991	5009	5010	5011	5016	5017	5029	5030	5031	5032	5033	5034	5035
5036	5038	5039	5040	5041	5055	5056	5057	5058	5059	5060	5061	5062
5064	5065	5066	5067	5079	5080	5081	5082	5083	5084	5085	5086	5088
5089	5091	5092	5110	5111	5112	5117	5118	5130	5131	5132	5133	5134
5135	5136	5137	5139	5140	5141	5142	5156	5157	5158	5159	5160	5161
5162	5163	5165	5166	5167	5168	5180	5181	5182	5183	5184	5185	5186
5187	5189	5190	5192	5193	5212	5213	5214	5219	5220	5232	5233	5234
5235	5236	5237	5238	5239	5241	5242	5243	5244	5258	5259	5260	5261
5262	5263	5264	5265	5267	5268	5269	5270	5282	5283	5284	5285	5286
5287	5288	5289	5291	5292	5294	5295	5313	5314	5315	5320	5321	5333
5334	5335	5336	5337	5338	5339	5340	5342	5343	5344	5345	5359	5360
5361	5362	5363	5364	5365	5366	5368	5369	5370	5371	5383	5384	5385
5386	5387	5388	5389	5390	5392	5393	5395	5396	5412	5413	5414	5456
5457	5458	5459	5460	5465	5466	5467	5473	5474	5493	5494	5495	5527
5528	5529	5530	5531	5535	5536	5537	5545	5546	5588	5589	5590	5645
5646	5647	5648	5649	5680	5681	5682	5728	5729	5730	5731	5732	5736
5737	5776	5777	5778	5833	5834	5835	5836	5837	5868	5869	5870	5878

```
                    5879  5893  5894  5895  5908  5909  5910  5911  5912  5914  5915  5916  5927
                    5928  5929  5930  5931  5933  5934  5935  5944  5945  5946  5947  5948  5951
                    5952  5967  5968  5969  5982  5983  5984  5985  5986  5988  5989  5990  6000
                    6001  6002  6003  6004  6006  6007  6008  6017  6018  6019  6020  6021  6024
                    6025  6054  6055  6056  6057  6058  6071  6072  6073  6074  6075  6087  6088
                    6089  6090  6091  6097  6098  6113  6114  6130  6131  6132  6133  6134  6152
                    6153  6154  6155  6156  6165  6166  6167  6168  6169  6173  6174  6175  6182
                    6183  6184  6185  6186  6195  6196  6197  6198  6199  6207  6208  6333  6334
                    6335  6336  6337  6338  6339  6340  6364  6365  6366  6367  6368  6386  6387
                    6388  6389  6390  6391  6392  6393  6412  6413  6414  6415  6416  6417  6418
                    6419  6423  6424  6425  6436  6437  6484  6485  6486  6487  6488  6492  6493
                    6508  6509  6510  6539  6540  6541  6542  6543  6546  6547  6548  6552  6553
                    6554  6561  6562  6579  6580  6581  6629  6630  6631  6632  6633  6635  6636
                    6637  6644  6645  6646  6647  6648  6651  6652  6653  6659  6660  6661  6662
                    6663  6665  6666  6667  6673  6674  6675  6676  6677  6679  6680  6681  6689
                    6690  6691  6692  6693  6695  6696  6697  6701  6702  6703  6704  6705  6706
                    6707  6708  6713  6714  6715  6716  6717  6718  6719  6720  6725  6726  6727
                    6728  6729  6731  6732  6733  6753  6754  6788  6789  6790  6807  6808  6809
                    6810  6811  6815  6816  6838  6839  6840  6841  6842  6843  6844  6845  6852
                    6853  6854  6855  6856  6861  6862  6878  6879  6880  6908  6909  6910  6911
                    6912  6915  6916  6917  6924  6925  6944  6945  6946  6966  6967  6968  6969
                    6970  6975  6976  7013  7014  7015  7016  7017  7021  7022  7041  7042  7043
                    7079  7080  7081  7082  7083  7088  7089  7108  7109  7110  7132  7133  7134
                    7135  7136  7137  7138  7140  7141  7158  7159  7160  7161  7162  7164  7165
                    7181  7182  7183  7184  7185  7188  7189  7191  7192  7204  7205  7206  7207
                    7208  7215  7216  7230  7231  7234  7235  7236  7237  7238  7239  7240  7241
                    7242  7243  7244  7251  7252  7253  7254  7255  7256  7257  7258  7342  7343
                    7347  7348  7358  7359  7360  7361
SVCSUB= 000C00       1250#  1262#
SVCTAG= 000000       1250#  1264#  1463  1464  1478  1479  2389  2390  2418  2419  2447  2448  2475
                    2476  2503  2504  2532  2533  2561  2562  2589  2590  2617  2618  2645  2646
                    2673  2674  2701  2702  2730  2731  2758  2759  2781  2782  2809  2810  2837
                    2838  2849  2850  2868  2869  2885  2886  2902  2903  2922  2923  2941  2942
                    2966  2967  2985  2986  3005  3006  3026  3027  3177  3178  3203  3204  3219
                    3220  3239  3240  3257  3258  3309  3310  3372  3373  3412  3413  3422  3423
                    3463  3464  3474  3475  3519  3520  3542  3543  3550  3551  3612  3613  3674
                    3675  3765  3766  3894  3895  3981  3982  4033  4034  4057  4058  4081  4082
                    4084  4085  4132  4133  4155  4156  4178  4179  4181  4182  4231  4232  4256
                    4257  4280  4281  4283  4284  4332  4333  4356  4357  4380  4381  4383  4384
                    4433  4434  4457  4458  4481  4482  4484  4485  4534  4535  4558  4559  4582
                    4583  4585  4586  4634  4635  4658  4659  4682  4683  4685  4686  4734  4735
                    4758  4759  4782  4783  4785  4786  4834  4835  4860  4861  4884  4885  4887
                    4888  4936  4937  4962  4963  4986  4987  4989  4990  5037  5038  5063  5064
                    5087  5088  5090  5091  5138  5139  5164  5165  5188  5189  5191  5192  5240
                    5241  5266  5267  5290  5291  5293  5294  5341  5342  5367  5368  5391  5392
                    5394  5395  5472  5473  5544  5545  5735  5736  5877  5878  5950  5951  6023
                    6024  6096  6097  6206  6207  6435  6436  6491  6492  6560  6561  6752  6753
                    6814  6815  6860  6861  6923  6924  6974  6975  7020  7021  7087  7088  7214
                    7215  7258  7259  7348  7349
SVCTST= 000000       1250#  1261#  3280  3281  3323  3324  3383  3384  3433  3434  3486  3487  3562
                    3563  3623  3624  3686  3687  3777  3778  3907  3908  3998  3999  4101  4102
                    4198  4199  4300  4301  4400  4401  4501  4502  4602  4603  4702  4703  4802
                    4803  4904  4905  5006  5007  5107  5108  5209  5210  5310  5311  5409  5410
                    5490  5491  5583  5584  5771  5772  5890  5891  5964  5965  6035  6036  6111
                    6112  6221  6222  6449  6450  6504  6505  6574  6575  6767  6768  6828  6829
                    6874  6875  6940  6941  6991  6992  7037  7038  7103  7104
```

```
                4727#   4751#   4775#   4827#   4853#   4877#   4929#   4955#   4979#   5030#   5056#   5080#   5131#
                5157#   5181#   5233#   5259#   5283#   5334#   5360#   5384#   5457#   5528#   5646#   5729#   5834#
                5909#   5928#   5945#   5983#   6001#   6018#   6055#   6072#   6088#   6131#   6153#   6166#   6183#
                6196#   6334#   6365#   6387#   6413#   6485#   6540#   6630#   6645#   6660#   6674#   6690#   6702#
                6714#   6726#   6808#   6839#   6853#   6909#   6967#   7014#   7080#   7135#   7159#   7182#   7205#
T$EXCP= 000000  7234#   7239    7240#   7244    7251#   7256
T$FLAG= 000040  3021#   3023    3288#   3305#   3351#   3386#   3409#   3436#   3460#   3489#   3516#   3539#   3599#
                3661#   3723#   3756#   3780#   3849#   3910#   3938#   4002#   4030#   4054#   4078#   4104#   4129#
                4152#   4175#   4201#   4228#   4253#   4277#   4303#   4329#   4353#   4377#   4403#   4430#   4454#
                4478#   4504#   4531#   4555#   4579#   4605#   4631#   4655#   4679#   4705#   4731#   4755#   4779#
                4805#   4831#   4857#   4881#   4907#   4933#   4959#   4983#   5009#   5034#   5060#   5084#   5110#
                5135#   5161#   5185#   5212#   5237#   5263#   5287#   5313#   5338#   5364#   5388#   5412#   5465#
                5493#   5535#   5588#   5680#   5776#   5868#   5893#   5914#   5933#   5967#   5988#   6006#   6173#
                6338#   6391#   6417#   6423#   6508#   6546#   6552#   6579#   6635#   6651#   6665#   6679#   6695#
                6706#   6718#   6731#   6788#   6843#   6878#   6915#   6944#   7041#
T$GMAN= 000000  1250#
T$HILI= 000001  7234#   7238    7240#   7243    7251#   7255
T$LAST= 000001  1250#   7359#
T$LOLI= 000000  7234#   7237    7240#   7242    7251#   7254
T$LSYM= 010000  1250#   1464    1479    2390    2419    2448    2476    2504    2533    2562    2590    2618    2646
                2674    2702    2731    2759    2782    2810    2838    2850    2869    2886    2903    2923    2942
                2967    2986    3006    3027    3178    3204    3220    3240    3258    3310    3373    3423    3475
                3551    3613    3675    3766    3895    3982    4085    4182    4284    4384    4485    4586    4686
                4786    4888    4990    5091    5192    5294    5395    5473    5545    5736    5878    5951    6024
                6097    6207    6436    6492    6561    6753    6815    6861    6924    6975    7021    7088    7215
                7259    7349
T$LTNO= 000053  7362#
T$NEST= 000000  1250#   1256#   1370#   1375#   1445#   1463#   1472#   1478#   2363#   2389#   2392#   2418#   2421#
                2447#   2450#   2475#   2478#   2503#   2506#   2532#   2535#   2561#   2564#   2589#   2592#   2617#
                2620#   2645#   2648#   2673#   2676#   2701#   2704#   2730#   2733#   2758#   2763#   2781#   2784#
                2809#   2812#   2837#   2841#   2849#   2853#   2868#   2871#   2885#   2888#   2902#   2906#   2922#
                2925#   2941#   2944#   2966#   2969#   2985#   2989#   3005#   3017#   3026#   3041#   3177#   3182#
                3203#   3214#   3219#   3234#   3239#   3255#   3257#   3281#   3309#   3324#   3372#   3384#   3395#
                3412#   3422#   3434#   3445#   3463#   3474#   3487#   3497#   3519#   3522#   3542#   3550#   3563#
                3612#   3624#   3674#   3687#   3765#   3778#   3894#   3908#   3981#   3999#   4010#   4033#   4036#
                4057#   4060#   4081#   4084#   4102#   4112#   4132#   4135#   4155#   4158#   4178#   4181#   4199#
                4209#   4231#   4234#   4256#   4259#   4280#   4283#   4301#   4311#   4332#   4335#   4356#   4359#
                4380#   4383#   4401#   4412#   4433#   4436#   4457#   4460#   4481#   4484#   4502#   4513#   4534#
                4537#   4558#   4561#   4582#   4585#   4603#   4613#   4634#   4637#   4658#   4661#   4682#   4685#
                4703#   4713#   4734#   4737#   4758#   4761#   4782#   4785#   4803#   4812#   4834#   4837#   4860#
                4863#   4884#   4887#   4905#   4914#   4936#   4939#   4962#   4965#   4986#   4989#   5007#   5016#
                5037#   5040#   5063#   5066#   5087#   5090#   5108#   5117#   5138#   5141#   5164#   5167#   5188#
                5191#   5210#   5219#   5240#   5243#   5266#   5269#   5290#   5293#   5311#   5320#   5341#   5344#
                5367#   5370#   5391#   5394#   5410#   5472#   5491#   5544#   5584#   5735#   5772#   5877#   5891#
                5950#   5965#   6023#   6036#   6096#   6112#   6206#   6222#   6435#   6450#   6491#   6505#   6560#
                6575#   6752#   6768#   6814#   6829#   6860#   6875#   6923#   6941#   6974#   6992#   7020#   7038#
                7087#   7104#   7214#   7230#   7257#   7342#   7347#
T$NSO = 000000  1256#
T$NS1 = 000005  1370#   1375    1445#   1463    1472#   1478    2363#   2389    2392#   2418    2421#   2447    2450#
                2475    2478#   2503    2506#   2532    2535#   2561    2564#   2589    2592#   2617    2620#   2645
                2648#   2673    2676#   2701    2704#   2730    2733#   2758    2763#   2781    2784#   2809    2812#
                2837    2841#   2849    2853#   2868    2871#   2885    2888#   2902    2906#   2922    2925#   2941
                2944#   2966    2969#   2985    2989#   3005    3017#   3026    3041#   3177    3182#   3203    3214#
                3219    3234#   3239    3255#   3257    3281#   3309    3324#   3372    3384#   3422    3434#   3474
                3487#   3550    3563#   3612    3624#   3674    3687#   3765    3778#   3894    3908#   3981    3999#
                4084    4102#   4181    4199#   4283    4301#   4383    4401#   4484    4502#   4585    4603#   4685
```

```
                    4703#  4785   4803#  4887   4905#  4989   5007#  5090   5108#  5191   5210#  5293   5311#
                    5394   5410#  5472   5491#  5544   5584#  5735   5772#  5877   5891#  5950   5965#  6023
                    6036#  6096   6112#  6206   6222#  6435   6450#  6491   6505#  6560   6575#  6752   6768#
                    6814   6829#  6860   6875#  6923   6941#  6974   6992#  7020   7038#  7087   7104#  7214
                    7230#  7257   7342#  7347
T$NS2 = 000003      3395#  3412   3445#  3463   3497#  3519   3522#  3542   4010#  4033   4036#  4057   4060#
                    4081   4112#  4132   4135#  4155   4158#  4178   4209#  4231   4234#  4256   4259#  4280
                    4311#  4332   4335#  4356   4359#  4380   4412#  4433   4436#  4457   4460#  4481   4513#
                    4534   4537#  4558   4561#  4582   4613#  4634   4637#  4658   4661#  4682   4713#  4734
                    4737#  4758   4761#  4782   4812#  4834   4837#  4860   4863#  4884   4914#  4936   4939#
                    4962   4965#  4986   5016#  5037   5040#  5063   5066#  5087   5117#  5138   5141#  5164
                    5167#  5188   5219#  5240   5243#  5266   5269#  5290   5320#  5341   5344#  5367   5370#
                    5391
T$PTNU= 000000      1250#
T$SAVL= 177777      1250#
T$SEGL= 177777      1250#  3395#  3410   3412#  3414   3445#  3461   3463#  3465   3497#  3517   3519#  3521
                    3522#  3540   3542#  3544   4010#  4031   4033#  4035   4036#  4055   4057#  4059   4060#
                    4079   4081#  4083   4112#  4130   4132#  4134   4135#  4153   4155#  4157   4158#  4176
                    4178#  4180   4209#  4229   4231#  4233   4234#  4254   4256#  4258   4259#  4278   4280#
                    4282   4311#  4330   4332#  4334   4335#  4354   4356#  4358   4359#  4378   4380#  4382
                    4412#  4431   4433#  4435   4436#  4455   4457#  4459   4460#  4479   4481#  4483   4513#
                    4532   4534#  4536   4537#  4556   4558#  4560   4561#  4580   4582#  4584   4613#  4632
                    4634#  4636   4637#  4656   4658#  4660   4661#  4680   4682#  4684   4713#  4732   4734#
                    4736   4737#  4756   4758#  4760   4761#  4780   4782#  4784   4812#  4832   4834#  4836
                    4837#  4858   4860#  4862   4863#  4882   4884#  4886   4914#  4934   4936#  4938   4939#
                    4960   4962#  4964   4965#  4984   4986#  4988   5016#  5035   5037#  5039   5040#  5061
                    5063#  5065   5066#  5085   5087#  5089   5117#  5136   5138#  5140   5141#  5162   5164#
                    5166   5167#  5186   5188#  5190   5219#  5238   5240#  5242   5243#  5264   5266#  5268
                    5269#  5288   5290#  5292   5320#  5339   5341#  5343   5344#  5365   5367#  5369   5370#
                    5389   5391#  5393
T$SEK0= 010002      3395#  3410   3412   3445#  3461   3463   3497#  3517   3519   3522#  3540   3542   4010#
                    4031   4033   4036#  4055   4057   4060#  4079   4081   4112#  4130   4132   4135#  4153
                    4155   4158#  4176   4178   4209#  4229   4231   4234#  4254   4256   4259#  4278   4280
                    4311#  4330   4332   4335#  4354   4356   4359#  4378   4380   4412#  4431   4433   4436#
                    4455   4457   4460#  4479   4481   4513#  4532   4534   4537#  4556   4558   4561#  4580
                    4582   4613#  4632   4634   4637#  4656   4658   4661#  4680   4682   4713#  4732   4734
                    4737#  4756   4758   4761#  4780   4782   4812#  4832   4834   4837#  4858   4860   4863#
                    4882   4884   4914#  4934   4936   4939#  4960   4962   4965#  4984   4986   5016#  5035
                    5037   5040#  5061   5063   5066#  5085   5087   5117#  5136   5138   5141#  5162   5164
                    5167#  5186   5188   5219#  5238   5240   5243#  5264   5266   5269#  5288   5290   5320#
                    5339   5341   5344#  5365   5367   5370#  5389   5391
T$SUBN= 000000      1250#  3280#  3323#  3383#  3433#  3486#  3562#  3623#  3686#  3777#  3907#  3998#  4101#
                    4198#  4300#  4400#  4501#  4602#  4702#  4802#  4904#  5006#  5107#  5209#  5310#  5409#
                    5490#  5583#  5771#  5890#  5964#  6035#  6111#  6221#  6449#  6504#  6574#  6767#  6828#
                    6874#  6940#  6991#  7037#  7103#
T$TAGL= 177777      1250#
T$TAGN= 010120      1250#  1370#  1445#  1472#  2363#  2392#  2421#  2450#  2478#  2506#  2535#  2564#  2592#
                    2620#  2648#  2676#  2704#  2733#  2763#  2784#  2812#  2841#  2853#  2871#  2888#  2906#
                    2925#  2944#  2969#  2989#  3017#  3041#  3182#  3214#  3234#  3255#  3281#  3324#  3384#
                    3434#  3487#  3563#  3624#  3687#  3778#  3908#  3999#  4102#  4199#  4301#  4401#  4502#
                    4603#  4703#  4803#  4905#  5007#  5108#  5210#  5311#  5410#  5491#  5584#  5772#  5891#
                    5965#  6036#  6112#  6222#  6450#  6505#  6575#  6768#  6829#  6875#  6941#  6992#  7038#
                    7104#  7230#  7342#
T$TEMP= 000005      1375#  1386#  1387#  1388#  1389#  1390#  1391#  1392#  1393#  1394#  1395#  1396#  1397#
                    1398#  1399#  1400#  1401#  1402#  1403#  1404#  1405#  1406#  1407#  1408#  1409#  1410#
                    1411#  1412#  1413#  1414#  1415#  1416#  1417#  1418#  1419#  1420#  1421#  1422#  1423#
```

```
                    4272  4277  4281  4284  4303  4311  4324  4329  4333  4335  4348  4353  4357
                    4359  4372  4377  4381  4384  4403  4412  4425  4430  4434  4436  4449  4454
                    4458  4460  4473  4478  4482  4485  4504  4513  4526  4531  4535  4537  4550
                    4555  4559  4561  4574  4579  4583  4586  4605  4613  4626  4631  4635  4637
                    4650  4655  4659  4661  4674  4679  4683  4686  4705  4713  4726  4731  4735
                    4737  4750  4755  4759  4761  4774  4779  4783  4786  4805  4812  4826  4831
                    4835  4837  4852  4857  4861  4863  4876  4881  4885  4888  4907  4914  4928
                    4933  4937  4939  4954  4959  4963  4965  4978  4983  4987  4990  5009  5016
                    5029  5034  5038  5040  5055  5060  5064  5066  5079  5084  5088  5091  5110
                    5117  5130  5135  5139  5141  5156  5161  5165  5167  5180  5185  5189  5192
                    5212  5219  5232  5237  5241  5243  5258  5263  5267  5269  5282  5287  5291
                    5294  5313  5320  5333  5338  5342  5344  5359  5364  5368  5370  5383  5388
                    5392  5395  5412  5456  5465  5473  5493  5527  5535  5545  5588  5645  5680
                    5728  5736  5776  5833  5868  5878  5893  5908  5914  5927  5933  5944  5951
                    5967  5982  5988  6000  6006  6017  6024  6054  6071  6087  6097  6113  6130
                    6152  6165  6173  6182  6195  6207  6333  6338  6364  6386  6391  6412  6417
                    6423  6436  6484  6492  6508  6539  6546  6552  6561  6579  6629  6635  6644
                    6651  6659  6665  6673  6679  6689  6695  6701  6706  6713  6718  6725  6731
                    6753  6788  6807  6815  6838  6843  6852  6861  6878  6908  6915  6924  6944
                    6966  6975  7013  7021  7041  7079  7088  7109  7132  7134  7140  7158  7164
                    7181  7188  7191  7204  7215
T$TSTS= 000001       1250#  3281#  3324#  3384#  3434#  3487#  3563#  3624#  3687#  3778#  3908#  3999#  4102#
                    4199#  4301#  4401#  4502#  4603#  4703#  4803#  4905#  5007#  5108#  5210#  5311#  5410#
                    5491#  5584#  5772#  5891#  5965#  6036#  6112#  6222#  6450#  6505#  6575#  6768#  6829#
                    6875#  6941#  6992#  7038#  7104#
T$$AU  = 010042      3255#  3257
T$$AUT= 010037       3182#  3203
T$$CLE= 010040       3214#  3219
T$$DU  = 010041      3234#  3239
T$$HAR= 010116       7230#  7258
T$$HW  = 010001      1445#  1463
T$$INI= 010036       3041#  3177
T$$MSG= 010034       2363#  2389  2392#  2418  2421#  2447  2450#  2475  2478#  2503  2506#  2532  2535#
                    2561  2564#  2589  2592#  2617  2620#  2645  2648#  2673  2676#  2701  2704#  2730
                    2733#  2758  2763#  2781  2784#  2809  2812#  2837  2841#  2849  2853#  2868  2871#
                    28    2888#  2902  2906#  2922  2925#  2941  2944#  2966  2969#  2985  2989#  3005
T$$PRO= 010000       1370#
T$$RPT= 010035       3017#  3021  3026
T$$SEG= 010002       3395#  3409  3412#  3445#  3460  3463#  3497#  3516  3519#  3522#  3539  3542#  4010#
                    4030  4033#  4036#  4054  4057#  4060#  4078  4081#  4112#  4129  4132#  4135#  4152
                    4155#  4158#  4175  4178#  4209#  4228  4231#  4234#  4253  4256#  4259#  4277  4280#
                    4311#  4329  4332#  4335#  4353  4356#  4359#  4377  4380#  4412#  4430  4433#  4436#
                    4454  4457#  4460#  4478  4481#  4513#  4531  4534#  4537#  4555  4558#  4561#  4579
                    4582#  4613#  4631  4634#  4637#  4655  4658#  4661#  4679  4682#  4713#  4731  4734#
                    4737#  4755  4758#  4761#  4779  4782#  4812#  4831  4834#  4837#  4857  4860#  4863#
                    4881  4884#  4914#  4933  4936#  4939#  4959  4962#  4965#  4983  4986#  5016#  5034
                    5037#  5040#  5060  5063#  5066#  5084  5087#  5117#  5135  5138#  5141#  5161  5164#
                    5167#  5185  5188#  5219#  5237  5240#  5243#  5263  5266#  5269#  5287  5290#  5320#
                    5338  5341#  5344#  5364  5367#  5370#  5388  5391#
T$$SOF= 010117       7342#  7348
T$$SW  = 010002      1472#  1478
T$$TES= 010115       3281#  3288  3305  3309  3324#  3351  3372  3384#  3386  3422  3434#  3436  3474
                    3487#  3489  3550  3563#  3599  3612  3624#  3661  3674  3687#  3723  3756  3765
                    3778#  3780  3849  3894  3908#  3910  3938  3981  3999#  4002  4084  4102#  4104
                    4181  4199#  4201  4283  4301#  4303  4383  4401#  4403  4484  4502#  4504  4585
                    4603#  4605  4685  4703#  4705  4785  4803#  4805  4887  4905#  4907  4989  5007#
```

```
                        5009    5090    5108#   5110    5191    5210#   5212    5293    5311#   5313    5394    5410#   5412
                        5465    5472    5491#   5493    5535    5544    5584#   5588    5680    5735    5772#   5776    5868
                        5877    5891#   5893    5914    5933    5950    5965#   5967    5988    6006    6023    6036#   6096
                        6112#   6173    6206    6222#   6338    6391    6417    6423    6435    6450#   6491    6505#   6508
                        6546    6552    6560    6575#   6579    6635    6651    6665    6679    6695    6706    6718    6731
                        6752    6768#   6788    6814    6829#   6843    6860    6875#   6878    6915    6923    6941#   6944
                        6974    6992#   7020    7038#   7041    7087    7104#   7214
T1          012146 G    1386    3280#
T10         014470 G    1395    3907#
T11         014736 G    1396    3998#
T12         015202 G    1397    4101#
T13         015432 G    1398    4198#
T14         015676 G    1399    4300#
T15         016142 G    1400    4400#
T16         016406 G    1401    4501#
T17         016652 G    1402    4602#
T18         017116 G    1403    4702#
T19         017362 G    1404    4802#
T2          012256 G    1387    3323#
T20         017642 G    1405    4904#
T21         020122 G    1406    5006#
T22         020376 G    1407    5107#
T23         020654 G    1408    5209#
T24         021130 G    1409    5310#
T25         021404 G    1410    5409#
T26         021576 G    1411    5490#
T27         021744 G    1412    5583#
T28         022322 G    1413    5771#
T29         022560 G    1414    5890#
T3          012422 G    1388    3383#
T30         022774 G    1415    5964#
T31         023210 G    1416    6035#
T32         023452 G    1417    6111#
T33         024044 G    1418    6221#
T34         025116 G    1419    6449#
T35         025216 G    1420    6504#
T36         025364 G    1421    6574#
T37         026042 G    1422    6767#
T38         026202 G    1423    6828#
T39         026310 G    1424    6874#
T4          012552 G    1389    3433#
T40         026446 G    1425    6940#
T41         026544 G    1426    6991#
T42         026644 G    1427    7037#
T43         026770 G    1428    7103#
T5          012712 G    1390    3486#
T6          013154 G    1391    3562#
T7          013356 G    1392    3623#
T8          013570 G    1393    3686#
T9          014112 G    1394    3777#
UAM     =   000200 G    1564#
VECTOR      027504      7272#
WPM         027400      7235    7260#
WROM        003422      2004#
WTYPE       002414      1631#   1887    1986    1988    2011    2033    3097*   3163    3165    3170    3505    3862    4001
                        4104    4201    4303    4403    4504    4605    4705    4805    4840    4907    4942    5009    5043
```

```
                    5110    5144    5212    5246    5313    5347    5638    5693    5826    6077    6507    6578    6877
                    6943    7040
X$ALWA= 000000      1250#
X$FALS= 000040      1250#
X$OFFS= 000400      1250#
X$TRUE= 000020      1250#
ZERO    002370      1621#
$BDADR  002450      1645#
$BDDAT  002454      1647#
$GDADR  002446      1644#
$GDDAT  002452      1646#   2972    6049*   6050    6052    6322*   6327    6330*   6331    6355*   6360    6362    6377*
                    6382    6384    6403*   6408    6410    7126*   7129    7153*   7176*   7177    7201*
$LSTIN= 000000      1258#
$LSTTA= 000000      1259#
$REG0   002430      1637#   2070*   2376    2405
$REG1   002426      1636#   2069*
$REG2   002424      1635#   2068*   2378    2407    2434    2548    2606    2690
$REG3   002422      1634#   2067*
$REG4   002420      1633#   2066*   2377    2406    2435    2463    2491    2520    2549    2577    2605    2633    2661
                    2689    2717    2746    2797    2825
$REG5   002416      1632#   2065*   2436    2464    2492    2521    2550    2578    2634    2662    2718    2747    2798
                    2826
$TEMP0  002442      1642#
$TMP0   002444      1643#   6040*   6065*   6066    6069    6080    6119*   6149    6156
.       = 030144    1241#   1624#   1625#   1626#   1627#   1660#   1706#   1732#   1870#   3022    3289    3306    3352
                    3387    3410    3437    3461    3490    3517    3540    3600    3662    3724    3757    3781    3850
                    3911    3926    3939    4003    4031    4055    4079    4105    4130    4153    4176    4202    4229
                    4254    4278    4304    4330    4354    4378    4404    4431    4455    4479    4505    4532    4556
                    4580    4606    4632    4656    4680    4706    4732    4756    4780    4806    4832    4858    4882
                    4908    4934    4960    4984    5010    5035    5061    5085    5111    5136    5162    5186    5202
                    5213    5238    5264    5288    5314    5339    5365    5389    5413    5466    5494    5536    5589
                    5681    5777    5869    5894    5915    5934    5968    5989    6007    6174    6339    6392    6418
                    6424    6428#   6432#   6509    6547    6553    6580    6636    6652    6666    6680    6696    6707
                    6719    6732    6789    6844    6879    6916    6945    7042    7322#
.MSTCL  002756      1865#   3326    3441    3495    3567    3628    3691    3786    3916    4009    4110    4207    4309
                    4409    4510    4610    4711    4810    4912    5014    5115    5217    5318    5417    5498    5591
                    5779    5896    5970    6039    6118    6129    6181    6194    6204    6225    6884    6950    7047
                    7107
.REGT   003050      1879#   1888
.ROMCL  003044      1876#   1905    1908    1911    1921    1929    1937    1945    1953    1956    1959    1970    1993
                    1999    3331    3334    3337    3355    3358    3574    3577    3581    3584    3587    3636    3639
                    3643    3646    3649    3698    3701    3705    3708    3711    3735    3738    3741    3744    3790
                    3793    3796    3803    3806    3825    3829    3869    3872    3875    3878    3918    3922    3942
                    3948    3961    3964    3967    5421    5424    5429    5445    5448    5507    5510    5519    5613
                    5616    5623    5626    5705    5708    5711    5715    5801    5804    5811    5814    5900    5918
                    5937    5974    5992    6010    6060    6230    6244    6248    6252    6266    6270    6274    6281
                    6284    6288    6292    6296    6305    6308    6324    6357    6375    6379    6398    6401    6405
                    6453    6456    6459    6462    6467    6471    6885    6888    6896    6900    6952    6956    7000
                    7007    7055    7059    7062    7072    7113    7117    7120    7124    7146    7150    7169    7172
                    7196
.SROMC  003100      1885#   1914    1962    1973    3809    3821    4015    4018    4039    4042    4063    4066    4114
                    4117    4137    4140    4160    4163    4212    4215    4237    4240    4262    4265    4314    4317
                    4338    4341    4362    4365    4415    4418    4439    4442    4463    4466    4516    4519    4540
                    4543    4564    4567    4616    4619    4640    4643    4664    4667    4716    4719    4740    4743
                    4764    4767    4816    4819    4842    4845    4866    4869    4918    4921    4944    4947    4968
                    4971    5019    5022    5045    5048    5069    5072    5120    5123    5146    5149    5170    5173
```

CROSS REFERENCE TABLE -- USER SYMBOLS

5222	5225	5248	5251	5272	5275	5323	5326	5349	5352	5373	5376	5513
5619	5807	6891										

BADHEA 1785# 3272 3276 3314 3319 3375 3379 3425 3429 3477 3482 3554 5558 3615 3619
3677 3682 3768 3773 3897 3903 3984 3994 4087 4097 4184 4194 4286 4296 4386
4396 4487 4497 4588 4598 4688 4698 4788 4798 4890 4900 4992 5002 5093 5103
5194 5205 5296 5306 5397 5405 5476 5486 5547 5579 5737 5767 5879 5886 5953
5960 6025 6031 6099 6107 6209 6217 6438 6445 6494 6500 6564 6570 6755 6763
6817 6824 6863 6870 6926 6936 6977 6987 7023 7033 7090 7099
BCOMPL 1# 1250# 3059 3065 3071
BERROR 1# 1250#
BGNAU 1# 1250# 3254
BGNAUT 1# 1250# 3181
BGNCLN 1# 1250# 3213
BGNDU 1# 1250# 3233
BGNHRD 1# 1250# 7229
BGNHW 1# 1250# 1444
BGNINI 1# 1250# 3040
BGNMOD 1# 1250# 1255
BGNMSG 1# 1250# 2363 2392 2421 2450 2478 2506 2535 2564 2592 2620 2648 2676 2704
2733 2763 2784 2812 2840 2852 2870 2887 2905 2924 2943 2968 2988
BGNPRO 1# 1250# 1369
BGNPTA 1# 1250#
BGNRPT 1# 1250# 3016
BGNSEG 1# 1250# 3394 3444 3496 3521 4009 4035 4059 4111 4134 4157 4208 4233 4258
4310 4334 4358 4411 4435 4459 4512 4536 4560 4612 4636 4660 4712 4736 4760
4811 4836 4862 4913 4938 4964 5015 5039 5065 5116 5140 5166 5218 5242 5268
5319 5343 5369
BGNSET 1# 1250#
BGNSFT 1# 1250# 7341
BGNSRV 1# 1250#
BGNSUB 1# 1250#
BGNSW 1# 1250# 1471
BGNTST 1# 1250# 3279 3322 3382 3432 3485 3561 3622 3685 3776 3906 3997 4100 4197
4299 4399 4500 4601 4701 4801 4903 5005 5106 5208 5309 5408 5489 5582 5770
5889 5963 6034 6110 6220 6448 6503 6573 6766 6827 6873 6939 6990 7036 7102
BNCOMP 1# 1250# 3077 3095
BNERRO 1# 1250#
BREAK 1# 1250# 3605 3667 3758 3851
BRESET 1# 1250# 3215 3236 6112 7131 7190
CKLOOP 1# 1250# 7139 7163 7187
CLOCK 1# 1250#
CLOSE 1# 1250#
CLRMAR 1825# 3792 3805
CLRVEC 1# 1250#
COMMEN 1# 1250#
DELAY 1# 1250#
DESCRI 1# 1250# 1593
DEVTYP 1# 1250# 1726
DISPAT 1# 1250# 1383
DISPLA 1# 1250#
DOCLN 1# 1250#
DODU 1# 1250# 3196
DORPT 1# 1250#
ED$CAL 1780# 3273 3277 3315 3320 3376 3380 3426 3430 3478 3483 3555 3559 3616 3620
3678 3683 3769 3774 3898 3904 3985 3995 4088 4098 4185 4195 4287 4297 4387
4397 4488 4498 4589 4599 4689 4699 4789 4799 4891 4901 4993 5003 5094 5104
5195 5206 5297 5307 5398 5406 5477 5487 5548 5580 5738 5768 5880 5887 5954
5961 6026 6032 6100 6108 6210 6218 6439 6446 6495 6501 6565 6571 6756 6764

CZDMQEO M8207 STATIC DIAG #2 MACY11 30A(1052) 18-OCT-82 15:30 PAGE 196
CZDMQE.P11 30-SEP-82 15:35 CROSS REFERENCE TABLE -- MACRO NAMES

```
        6818    6825    6864    6871    6927    6937    6978    6988    7024    7034    7091    7100
ENDAU      1#   1250#   3256
ENDAUT     1#   1250#   3202
ENDCLN     1#   1250#   3218
ENDCOM     1#   1250#
ENDDU      1#   1250#   3238
ENDHRD     1#   1250#   7256
ENDHW      1#   1250#   1462
ENDINI     1#   1250#   3176
ENDMOD     1#   1250#
ENDMSG     1#   1250#   2389    2418    2447    2475    2503    2532    2561    2589    2617    2645    2673    2701    2730
        2758    2781    2809    2837    2848    2867    2884    2901    2921    2940    2965    2984    3004
ENDPRO     1#   1250#   1374
ENDPTA     1#   1250#
ENDRPT     1#   1250#   3025
ENDSEG     1#   1250#   3411    3462    3518    3541    4032    4056    4080    4131    4154    4177    4230    4255    4279
        4331    4355    4379    4432    4456    4480    4533    4557    4581    4633    4657    4681    4733    4757    4781
        4833    4859    4883    4935    4961    4985    5036    5062    5086    5137    5163    5187    5239    5265    5289
        5340    5366    5390
ENDSET     1#   1250#
ENDSFT     1#   1250#   7346
ENDSRV     1#   1250#
ENDSUB     1#   1250#
ENDSW      1#   1250#   1477
ENDTST     1#   1250#   3308    3371    3421    3473    3549    3611    3673    3764    3893    3980    4083    4180    4282
        4382    4483    4584    4684    4784    4886    4988    5089    5190    5292    5393    5471    5543    5734    5876
        5949    6022    6095    6205    6434    6490    6559    6751    6813    6859    6922    6973    7019    7086    7213
EQUALS     1#   1250#   1504
ERRDF      1#   1250#   3297    3345    3365    3404    3455    3511    3534    3594    3656    3718    3751    3816    3841
        3888    3933    3956    3975    4025    4049    4073    4124    4147    4170    4222    4247    4272    4324    4348
        4372    4425    4449    4473    4526    4550    4574    4626    4650    4674    4726    4750    4774    4826    4852
        4876    4928    4954    4978    5029    5055    5079    5130    5156    5180    5232    5258    5282    5333    5359
        5383    5456    5527    5645    5728    5833    5908    5927    5944    5982    6000    6017    6053    6070    6086
        6130    6152    6165    6182    6195    6333    6364    6386    6412    6484    6539    6629    6644    6659    6673
        6689    6701    6713    6725    6807    6838    6852    6908    6966    7013    7079    7134    7158    7181    7204
ERRHRD     1#   1250#
ERROR      1#   1250#   2355#   3296    3344    3364    3403    3454    3510    3533    3593    3655    3717    3750    3815
        3840    3887    3932    3955    3974    4024    4048    4072    4123    4146    4169    4221    4246    4271    4323
        4347    4371    4424    4448    4472    4525    4549    4573    4625    4649    4673    4725    4749    4773    4825
        4851    4875    4927    4953    4977    5028    5054    5078    5129    5155    5179    5231    5257    5281    5332
        5358    5382    5455    5526    5644    5727    5832    5907    5926    5943    5981    5999    6016    6129    6151
        6164    6181    6194    6332    6363    6385    6411    6483    6538    6628    6643    6658    6672    6688    6700
        6712    6724    6806    6837    6851    6907    6965    7012    7078    7133    7157    7180    7203
ERRSF      1#   1250#
ERRSOF     1#   1250#
ERRTBL     1#   1250#
ESCAPE     1#   1250#   3287    3304    3350    3408    3459    3515    3538    3598    3660    3722    3755    3848    3937
        4029    4053    4077    4128    4151    4174    4227    4252    4276    4328    4352    4376    4429    4453    4477
        4530    4554    4578    4630    4654    4678    4730    4754    4778    4830    4856    4880    4932    4958    4982
        5033    5059    5083    5134    5160    5184    5236    5262    5286    5337    5363    5387    5464    5534    5679
        5867    5913    5932    5987    6005    6337    6390    6416    6545    6634    6650    6664    6678    6694    6705
        6717    6842    6914
EXIT       1#   1250#   3020    3386    3436    3489    3780    3910    4001    4104    4201    4303    4403    4504    4605
        4705    4805    4907    5009    5110    5212    5313    5412    5493    5588    5776    5893    5967    6172    6422
        6507    6551    6578    6730    6787    6877    6943    7040
FEQUAL     1#   1250#
```

```
GETBYT      1#   1250#
GETPRI      1#   1250#
GETWOR      1#   1250#
GMANIA      1#   1250#
GMANID      1#   1250#
GMANIL      1#   1250#
GPHARD      1#   1250#  3091
GPRMA       1#   1250#  7239
GPRMD       1#   1250#  7233  7250
GPRML       1#   1250#
HEADER      1#   1250#  1281
INLOOP      1#   1250#
IOSETU      1#   1250#
IOSTAR      1#   1250#
KT11        1#   1250#
K4ONLY   1775#   1814#  3778  5410  5491  5586  5774  5891  5965
LASTAD      1#   1250#  7357
MACEX    1796#   3384   3434  3487  3908
MACEX2   1805#   4102   4199  4301  4401  4502  4603  4703  4803  4905  5007  5108  5210  5311
MANUAL      1#   1250#
MDT0     2333#   2776
MDT1     2313#   2376   2405
MDT2     2317#   2434
MDT3     2321#   2463   2491  2577  2661  2797  2825
MDT4     2325#   2519
MDT5     2329#   2548
MDT6     2335#   2605   2689
MDT7     2339#   2633   2746
MDT8     2342#   2717
MEMORY      1#   1250#
MSTCLR   1861#   3325   3440  3494  3566  3627  3690  3785  3915  4008  4109  4206  4308  4408  4509
         4609    4710   4809  4911  5013  5114  5216  5317  5416  5497  5590  5778  5895  5969  6038
         6117    6128   6180  6193  6203  6224  6883  6949  7046  7106
MYINT    1790#   3388   3438  3491  3563  3624  3687  3782  3912  4005  4106  4203  4305  4405  4506
         4607    4707   4807  4909  5011  5112  5214  5315  5414  5495  5584  5772  5896  5970  6036
         6114    6222   6450  6511  6582  6768  6829  6881  6947  6992  7044  7104
MSBYTE      1#   1250#  1282#  1288  1289  1290
MSCHEC      1#   1250#  3021#  3386#  3436#  3489#  3780#  3910#  4002#  4104#  4201#  4303#  4403#  4504#  4605#
         4705#   4805#  4907#  5009#  5110#  5212#  5313#  5412#  5493#  5588#  5776#  5893#  5967#  6173#  6423#
         6508#   6552#  6579#  6731#  6788#  6878#  6944#  7041#
MSCNTO      1#   1250#  7234#  7240#  7251#
MSCOUN      1#   1250#  2364#  2370#  2376#  2384#  2393#  2399#  2405#  2413#  2422#  2428#  2434#  2442#  2451#
         2457#   2463#  2470#  2479#  2485#  2491#  2498#  2507#  2513#  2519#  2527#  2536#  2542#  2548#  2556#
         2565#   2571#  2577#  2584#  2593#  2599#  2605#  2612#  2621#  2627#  2633#  2640#  2649#  2655#  2661#
         2668#   2677#  2683#  2689#  2696#  2705#  2711#  2717#  2725#  2734#  2740#  2746#  2753#  2764#  2770#
         2776#   2785#  2791#  2797#  2804#  2813#  2819#  2825#  2832#  2843#  2855#  2862#  2873#  2879#  2890#
         2896#   2908#  2916#  2927#  2935#  2946#  2954#  2960#  2971#  2979#  2991#  2999#
MSDATA      1#   1250#  1282#  1291  1293  1295  1297  1299  1301  1303  1305  1307  1309  1311  1313
         1315    1317   1319  1321#  1323  1325  1328  1331  1333  1335  1337  1339  1341  1343  1345
         1347    1349   1351  1353  1355  1357  1359  1361  1363  1365  1594#  1727#
MSDECR      1#   1250#  1375#  1463#  1478#  2389#  2418#  2447#  2475#  2503#  2532#  2561#  2589#  2617#  2645#
         2673#   2701#  2730#  2758#  2781#  2809#  2837#  2849#  2868#  2885#  2902#  2922#  2941#  2966#  2985#
         3005#   3026#  3177#  3203#  3219#  3239#  3257#  3309#  3372#  3412#  3422#  3463#  3474#  3519#  3542#
         3550#   3612#  3674#  3765#  3894#  3981#  4033#  4057#  4081#  4084#  4132#  4155#  4178#  4181#  4231#
         4256#   4280#  4283#  4332#  4356#  4380#  4383#  4433#  4457#  4481#  4484#  4534#  4558#  4582#  4585#
         4634#   4658#  4682#  4685#  4734#  4758#  4782#  4785#  4834#  4860#  4884#  4887#  4936#  4962#  4986#
```

```
                4989#  5037#  5063#  5087#  5090#  5138#  5164#  5188#  5191#  5240#  5266#  5290#  5293#  5341#  5367#
                5391#  5394#  5472#  5544#  5735#  5877#  5950#  6023#  6096#  6206#  6435#  6491#  6560#  6752#  6814#
                6860#  6923#  6974#  7020#  7087#  7214#  7257#  7347#
M$DEFA             1#  1250#  7234#  7240#  7251#
M$ENDE             1#  1250#  1463#  1478#  2389#  2418#  2447#  2475#  2503#  2532#  2561#  2589#  2617#  2645#  2673#
                2701#  2730#  2758#  2781#  2809#  2837#  2849#  2868#  2885#  2902#  2922#  2941#  2966#  2985#  3005#
                3026#  3177#  3203#  3219#  3239#  3257#  3309#  3372#  3412#  3422#  3463#  3474#  3519#  3542#  3550#
                3612#  3674#  3765#  3894#  3981#  4033#  4057#  4081#  4084#  4132#  4155#  4178#  4181#  4231#  4256#
                4280#  4283#  4332#  4356#  4380#  4383#  4433#  4457#  4481#  4484#  4534#  4558#  4582#  4585#  4634#
                4658#  4682#  4685#  4734#  4758#  4782#  4785#  4834#  4860#  4884#  4887#  4930#  4962#  4986#  4989#
                5037#  5063#  5087#  5090#  5138#  5164#  5188#  5191#  5240#  5266#  5290#  5293#  5341#  5367#  5391#
                5394#  5472#  5544#  5735#  5877#  5950#  6023#  6096#  6206#  6435#  6491#  6560#  6752#  6814#  6860#
                6923#  6974#  7020#  7087#  7214#  7257#  7347#
M$ERRI             1#  1250#  3297#  3345#  3365#  3404#  3455#  3511#  3534#  3594#  3656#  3718#  3751#  3816#  3841#
                3888#  3933#  3956#  3975#  4025#  4049#  4073#  4124#  4147#  4170#  4222#  4247#  4272#  4324#  4348#
                4372#  4425#  4449#  4473#  4526#  4550#  4574#  4626#  4650#  4674#  4726#  4750#  4774#  4826#  4852#
                4876#  4928#  4954#  4978#  5029#  5055#  5079#  5130#  5156#  5180#  5232#  5258#  5282#  5333#  5359#
                5383#  5456#  5527#  5645#  5728#  5833#  5908#  5927#  5944#  5982#  6000#  6017#  6054#  6071#  6087#
                6130#  6152#  6165#  6182#  6195#  6333#  6364#  6386#  6412#  6484#  6539#  6629#  6644#  6659#  6673#
                6689#  6701#  6713#  6725#  6807#  6838#  6852#  6908#  6966#  7013#  7079#  7134#  7158#  7181#  7204#
M$ESCA             1#  1250#  3288#  3289   3305#  3306   3351#  3352   3409#  3460#  3516#  3539#  3599#  3600   3661#
                3662   3723#  3724   3756#  3757   3849#  3850   3938#  3939   4030#  4054#  4078#  4129#  4152#  4175#
                4228#  4253#  4277#  4329#  4353#  4377#  4430#  4454#  4478#  4531#  4555#  4579#  4631#  4655#  4679#
                4731#  4755#  4779#  4831#  4857#  4881#  4933#  4959#  4983#  5034#  5060#  5084#  5135#  5161#  5185#
                5237#  5263#  5287#  5338#  5364#  5388#  5465#  5466   5535#  5536   5680#  5681   5868#  5869   5914#
                5915   5933#  5934   5988#  5989   6006#  6007   6338#  6339   6391#  6392   6417#  6418   6546#  6547
                6635#  6636   6651#  6652   6665#  6666   6679#  6680   6695#  6696   6706#  6707   6718#  6719   6843#
                6844   6915#  6916
M$ESCS             1#  1250#  3288#  3305#  3351#  3409#  3410   3460#  3461   3516#  3517   3539#  3540   3599#  3661#
                3723#  3756#  3849#  3938#  4030#  4031   4054#  4055   4078#  4079   4129#  4130   4152#  4153   4175#
                4176   4228#  4229   4253#  4254   4277#  4278   4329#  4330   4353#  4354   4377#  4378   4430#  4431
                4454#  4455   4478#  4479   4531#  4532   4555#  4556   4579#  4580   4631#  4632   4655#  4656   4679#
                4680   4731#  4732   4755#  4756   4779#  4780   4831#  4832   4857#  4858   4881#  4882   4933#  4934
                4959#  4960   4983#  4984   5034#  5035   5060#  5061   5084#  5085   5135#  5136   5161#  5162   5185#
                5186   5237#  5238   5263#  5264   5287#  5288   5338#  5339   5364#  5365   5388#  5389   5465#  5535#
                5680#  5868#  5914#  5933#  5988#  6006#  6338#  6391#  6417#  6546#  6635#  6651#  6665#  6679#  6695#
                6706#  6718#  6843#  6915#
M$EXCP             1#  1250#  7234#  7240#  7251#
M$EXIT             1#  1250#  3021#  3386#  3387   3436#  3437   3489#  3490   3780#  3781   3910#  3911   4002#  4003
                41J4#  4105   4201#  4202   4303#  4304   4403#  4404   4504#  4505   4605#  4606   4705#  4706   4805#
                4806   4907#  4908   5009#  5010   5110#  5111   5212#  5213   5313#  5314   5412#  5413   5493#  5494
                5588#  5589   5776#  5777   5893#  5894   5967#  5968   6173#  6174   6423#  6424   650E#  6509   6552#
                6553   6579#  6580   6731#  6732   6788#  6789   6878#  6879   6944#  6945   7041#  7042
M$EXSE             1#  1250#  3021#  3386#  3436#  3489#  3780#  3910#  4002#  4104#  4201#  4303#  4403#  4504#  4605#
                4705#  4805#  4907#  5009#  5110#  5212#  5313#  5412#  5493#  5588#  5776#  5893#  5967#  6173#  6423#
                6508#  6552#  6579#  6731#  6788#  6878#  6944#  7041#
M$EXTJ             1#  1250#  3021#  3022   3386#  3436#  3489#  3780#  3910#  4002#  4104#  4201#  4303#  4403#  4504#
                4605#  4705#  4805#  4907#  5009#  5110#  5212#  5313#  5412#  5493#  5588#  5776#  5893#  5967#  6173#
                6423#  6508#  6552#  6579#  6731#  6788#  6878#  6944#  7041#
M$GEN              1#  1250#  1256#  1282#  1291#  1293#  1295#  1297#  1299#  1301#  1303#  1305#  1307#  1309#  1311#
                1313#  1315#  1317#  1319#  1321#  1323#  1325#  1328#  1331#  1333#  1335#  1337#  1339#  1341#  1343#
                1345#  1347#  1349#  1351#  1353#  1355#  1357#  1359#  1361#  1363#  1365#  1370#  1385#  1446#  1447#
                1463#  1473#  1474#  1478#  1594#  1727#  2363#  2389#  2392#  2418#  2421#  2447#  2450#  2475#  2478#
                2503#  2506#  2532#  2535#  2561#  2564#  2589#  2592#  2617#  2620#  2645#  2648#  2673#  2676#  2701#
                2704#  2730#  2733#  2758#  2763#  2781#  2784#  2809#  2812#  2837#  2841#  2849#  2853#  2868#  2871#
                2885#  2888#  2902#  2906#  2922#  2925#  2941#  2944#  2966#  2969#  2985#  2989#  3005#  3017#  3026#
```

```
                3041#  3177#  3182#  3203#  3214#  3219#  3234#  3239#  3255#  3257#  3280#  3309#  3323#  3372#  3383#
                3412#  3422#  3433#  3463#  3474#  3486#  3519#  3542#  3550#  3562#  3612#  3623#  3674#  3686#  3765#
                3777#  3894#  3907#  3981#  3998#  4033#  4057#  4081#  4084#  4101#  4132#  4155#  4178#  4181#  4198#
                4231#  4256#  4280#  4283#  4300#  4332#  4356#  4380#  4383#  4400#  4433#  4457#  4481#  4484#  4501#
                4534#  4558#  4582#  4585#  4602#  4634#  4658#  4682#  4685#  4702#  4734#  4758#  4782#  4785#  4802#
                4834#  4860#  4884#  4887#  4904#  4936#  4962#  4986#  4989#  5006#  5037#  5063#  5087#  5090#  5107#
                5138#  5164#  5188#  5191#  5209#  5240#  5266#  5290#  5293#  5310#  5341#  5367#  5391#  5394#  5409#
                5472#  5490#  5544#  5583#  5735#  5771#  5877#  5890#  5950#  5964#  6023#  6035#  6096#  6111#  6206#
                6221#  6435#  6449#  6491#  6504#  6560#  6574#  6752#  6767#  6814#  6828#  6860#  6874#  6923#  6940#
                6974#  6991#  7020#  7037#  7087#  7103#  7214#  7231#  7258#  7343#  7348#  7361#
M$GENB             1#  1250#
M$GETS             1#  1250#  1375#  1463#  1478#  2389#  2418#  2447#  2475#  2503#  2532#  2561#  2589#  2617#  2645#
                2673#  2701#  2730#  2758#  2781#  2809#  2837#  2849#  2868#  2885#  2902#  2922#  2941#  2966#  2985#
                3005#  3026#  3177#  3203#  3219#  3239#  3257#  3309#  3372#  3410#  3412#  3422#  3461#  3463#  3474#
                3517#  3519#  3540#  3542#  3550#  3612#  3674#  3765#  3894#  3981#  4031#  4033#  4055#  4057#  4079#
                4081#  4084#  4130#  4132#  4153#  4155#  4176#  4178#  4181#  4229#  4231#  4254#  4256#  4278#  4280#
                4283#  4330#  4332#  4354#  4356#  4378#  4380#  4383#  4431#  4433#  4455#  4457#  4479#  4481#  4484#
                4532#  4534#  4556#  4558#  4580#  4582#  4585#  4632#  4634#  4656#  4658#  4680#  4682#  4685#  4732#
                4734#  4756#  4758#  4780#  4782#  4785#  4832#  4834#  4858#  4860#  4882#  4884#  4887#  4934#  4936#
                4960#  4962#  4984#  4986#  4989#  5035#  5037#  5061#  5063#  5085#  5087#  5090#  5136#  5138#  5162#
                5164#  5186#  5188#  5191#  5238#  5240#  5264#  5266#  5288#  5290#  5293#  5339#  5341#  5365#  5367#
                5389#  5391#  5394#  5472#  5544#  5735#  5877#  5950#  6023#  6096#  6206#  6435#  6491#  6560#  6752#
                6814#  6860#  6923#  6974#  7020#  7087#  7214#  7257#  7347#
M$GETT             1#  1250#  7021#  3288#  3305#  3351#  3386#  3409#  3410   3436#  3460#  3461   3489#  3516#  3517
                3539#  3540   3599#  3661#  3723#  3756#  3780#  3849#  3910#  3938#  4002#  4030#  4031   4054#  4055
                4078#  4079   4104#  4129#  4130   4152#  4153   4175#  4176   4201#  4228#  4229   4253#  4254   4277#
                4278   4303#  4329#  4330   4353#  4354   4377#  4378   4403#  4430#  4431   4454#  4455   4478#  4479
                4504#  4531#  4532   4555#  4556   4579#  4580   4605#  4631#  4632   4655#  4656   4679#  4680   4705#
                4731#  4732   4755#  4756   4779#  4780   4805#  4831#  4832   4857#  4858   4881#  4882   4907#  4933#
                4934   4959#  4960   4983#  4984   5009#  5034#  5035   5060#  5061   5084#  5085   5110#  5135#  5136
                5161#  5162   5185#  5186   5212#  5237#  5238   5263#  5264   5287#  5288   5313#  5338#  5339   5364#
                5365   5388#  5389   5412#  5465#  5493#  5535#  5588#  5680#  5776#  5868#  5893#  5914#  5933#  5967#
                5988#  6006#  6173#  6338#  6391#  6417#  6423#  6506#  6546#  6552#  6579#  6635#  6651#  6665#  6679#
                6695#  6706#  6718#  6731#  6788#  6843#  6878#  6915#  6944#  7041#
M$GNGB             1#  1250#  1256#  1282#  1291#  1293#  1295#  1297#  1299#  1301#  1303#  1305#  1307#  1309#  1311#
                1313#  1315#  1317#  1319#  1321#  1323#  1325#  1328#  1331#  1333#  1335#  1337#  1339#  1341#  1343#
                1345#  1347#  1349#  1351#  1353#  1355#  1357#  1359#  1361#  1363#  1365#  1370#  1384#  1385   1445#
                1446   1447   1472#  1473   1474   1594#  1727#  2363#  2392#  2421#  2450#  2478#  2506#  2535#  2564#
                2592#  2620#  2648#  2676#  2704#  2733#  2763#  2784#  2812#  2841#  2853#  2871#  2888#  2906#  2925#
                2944#  2969#  2989#  3017#  3041#  3182#  3214#  3234#  3255#  7230#  7231   7342#  7343   7358#  7361
M$GNIN             1#  1250#  1282#  1283   1284   1285   1286   1287   1288#  1289#  1290#  1291#  1292   1293#  1294
                1295#  1296   1297#  1298   1299#  1300   1301#  1302   1303#  1304   1305#  1306   1307#  1308   1309#
                1310   1311#  1312   1313#  1314   1315#  1316   1317#  1318   1319#  1320   1321#  1322   1323#  1324
                1325#  1326   1327   1328#  1329   1330#  1331#  1332   1333#  1334   1335#  1336   1337#  1338   1339#
                1340   1341#  1342   1343#  1344   1345#  1346   1347#  1348   1349#  1350   1351#  1352   1353#  1354
                1355#  1356   1357#  1358   1359#  1360   1361#  1362   1363#  1364   1365#  1366   1384#  1386#  1387#
                1388#  1389#  1390#  1391#  1392#  1393#  1394#  1395#  1396#  1397#  1398#  1399#  1400#  1401#  1402#
                1403#  1404#  1405#  1406#  1407#  1408#  1409#  1410#  1411#  1412#  1413#  1414#  1415#  1416#  1417#
                1418#  1419#  1420#  1421#  1422#  1423#  1424#  1425#  1426#  1427#  1428#  1445#  1472#  1594#  1595
                1599   1727#  1728   1732   2364#  2365#  2366#  2367   2368#  2369   2370#  2371#  2372#  2373   2374#
                2375   2376#  2377#  2378#  2379#  2380#  2381   2382#  2383   2384#  2385#  2386   2387#  2388   2390#
                2393#  2394#  2395#  2396   2397#  2398   2399#  2400#  2401#  2402   2403#  2404   2405#  2406#  2407#
                2408#  2409#  2410   2411#  2412   2413#  2414#  2415   2416#  2417   2419#  2422#  2423#  2424#  2425
                2426#  2427   2428#  2429#  2430#  2431   2432#  2433   2434#  2435#  2436#  2437#  2438#  2439   2440#
                2441   2442#  2443#  2444   2445#  2446   2448#  2451#  2452#  2453#  2454   2455#  2456   2457#  2458#
                2459#  2460   2461#  2462   2463#  2464#  2465#  2466#  2467   2468#  2469   2470#  2471#  2472   2473#
```

2474 2476# 2479# 2480# 2481# 2482 2483# 2484 2485# 2486# 2487# 2488 2489# 2490 2491#
2492# 2493# 2494# 2495 2496# 2497 2498# 2499# 2500 25[illegible]# 2502 2504# 2507# 2508# 2509#
2510 2511# 2512 2513# 2514# 2515# 2516 2517# 2518 2519# 2520# 2521# 2522# 2523# 2524
2525# 2526 2527# 2528# 2529 2530# 2531 2533# 2536# 2537# 2538# 2539 2540# 2541 2542#
2543# 2544# 2545 2546# 2547 2548# 2549# 2550# 2551# 2552# 2553 2554# 2555 2556# 2557#
2558 2559# 2560 2562# 2565# 2566# 2567# 2568 2569# 2570 2571# 2572# 2573# 2574 2575#
2576 2577# 2578# 2579# 2580# 2581 2582# 2583 2584# 2585# 2586 2587# 2588 2590# 2593#
2594# 2595# 2596 2597# 2598 2599# 2600# 2601# 2602 2603# 2604 2605# 2606# 2607# 2608#
2609 2610# 2611 2612# 2613# 2614 2615# 2616 2618# 2621# 2622# 2623# 2624 2625# 2626
2627# 2628# 2629# 2630 2631# 2632 2633# 2634# 2635# 2636# 2637 2638# 2639 2640# 2641#
2642 2643# 2644 2646# 2649# 2650# 2651# 2652 2653# 2654 2655# 2656# 2657# 2658 2659#
2660 2661# 2662# 2663# 2664# 2665 2666# 2667 2668# 2669# 2670 2671# 2672 2674# 2677#
2678# 2679# 2680 2681# 2682 2683# 2684# 2685# 2686 2687# 2688 2689# 2690# 2691# 2692#
2693 2694# 2695 2696# 2697# 2698 2699# 2700 2702# 2705# 2706# 2707# 2708 2709# 2710
2711# 2712# 2713# 2714 2715# 2716 2717# 2718# 2719# 2720# 2721# 2722 2723# 2724 2725#
2726# 2727 2728# 2729 2731# 2734# 2735# 2736# 2737 2738# 2739 2740# 2741# 2742# 2743
2744# 2745 2746# 2747# 2748# 2749# 2750 2751# 2752 2753# 2754# 2755 2756# 2757 2759#
2764# 2765# 2766# 2767 2768# 2769 2770# 2771# 2772# 2773 2774# 2775 2776# 2777# 2778
2779# 2780 2782# 2785# 2786# 2787# 2788 2789# 2790 2791# 2792# 2793# 2794 2795# 2796
2797# 2798# 2799# 2800# 2801 2802# 2803 2804# 2805# 2806 2807# 2808 2810# 2813# 2814#
2815# 2816 2817# 2818 2819# 2820# 2821# 2822 2823# 2824 2825# 2826# 2827# 2828# 2829
2830# 2831 2832# 2833# 2834 2835# 2836 2838# 2843# 2844# 2845 2846# 2847 2850# 2855#
2856# 2857# 2858 2859# 2860 2862# 2863# 2864 2865# 2866 2869# 2873# 2874# 2875 2876#
2877 2879# 2880# 2881 2882# 2883 2886# 2890# 2891# 2892 2893# 2894 2896# 2897# 2898
2899# 2900 2903# 2908# 2909# 2910# 2911# 2912 2913# 2914 2916# 2917# 2918 2919# 2920
2923# 2927# 2928# 2929# 2930# 2931 2932# 2933 2935# 2936# 2937 2938# 2939 2942# 2946#
2947# 2948# 2949# 2950 2951# 2952 2954# 2955# 2956 2957# 2958 2960# 2961# 2962 2963#
2964 2967# 2971# 2972# 2973# 2974# 2975 2976# 2977 2979# 2980# 2981 2982# 2983 2986#
2991# 2992# 2993# 2994# 2995 2996# 2997 2999# 3000# 3001 3002# 3003 3006# 3021# 3022#
3027# 3057# 3058# 3060# 3063# 3064# 3066# 3069# 3070# 3072# 3075# 3076# 3078# 3092# 3093#
3094# 3096# 3178# 3197# 3198# 3204# 3216# 3220# 3237# 3240# 3258# 3288# 3289# 3297# 3298#
3299# 3300# 3305# 3306# 3310# 3345# 3346# 3347# 3348# 3351# 3352# 3365# 3366# 3367# 3368#
3373# 3386# 3387# 3395# 3404# 3405# 3406# 3407# 3409# 3410# 3413# 3423# 3436# 3437# 3445#
3455# 3456# 3457# 3458# 3460# 3461# 3464# 3475# 3489# 3490# 3497# 3511# 3512# 3513# 3514#
3516# 3517# 3520# 3522# 3534# 3535# 3536# 3537# 3539# 3540# 3543# 3551# 3594# 3595# 3596#
3597# 3599# 3600# 3606# 3613# 3656# 3657# 3658# 3659# 3661# 3662# 3668# 3675# 3718# 3719#
3720# 3721# 3723# 3724# 3751# 3752# 3753# 3754# 3756# 3757# 3759# 3766# 3780# 3781# 3816#
3817# 3818# 3819# 3841# 3842# 3843# 3844# 3849# 3850# 3852# 3888# 3889# 3890# 3891# 3895#
3910# 3911# 3933# 3934# 3935# 3936# 3938# 3939# 3956# 3957# 3958# 3959# 3975# 3976# 3977#
3978# 3982# 4002# 4003# 4010# 4025# 4026# 4027# 4028# 4030# 4031# 4034# 4036# 4049# 4050#
4051# 4052# 4054# 4055# 4058# 4060# 4073# 4074# 4075# 4076# 4078# 4079# 4082# 4085# 4104#
4105# 4112# 4124# 4125# 4126# 4127# 4129# 4130# 4133# 4135# 4147# 4148# 4149# 4150# 4152#
4153# 4156# 4158# 4170# 4171# 4172# 4173# 4175# 4176# 4179# 4182# 4201# 4202# 4209# 4222#
4223# 4224# 4225# 4228# 4229# 4232# 4234# 4247# 4248# 4249# 4250# 4253# 4254# 4257# 4259#
4272# 4273# 4274# 4275# 4277# 4278# 4281# 4284# 4303# 4304# 4311# 4324# 4325# 4326# 4327#
4329# 4330# 4333# 4335# 4348# 4349# 4350# 4351# 4353# 4354# 4357# 4359# 4372# 4373# 4374#
4375# 4377# 4378# 4381# 4384# 4403# 4404# 4412# 4425# 4426# 4427# 4428# 4430# 4431# 4434#
4436# 4449# 4450# 4451# 4452# 4454# 4455# 4458# 4460# 4473# 4474# 4475# 4476# 4478# 4479#
4482# 4485# 4504# 4505# 4513# 4526# 4527# 4528# 4529# 4531# 4532# 4535# 4537# 4550# 4551#
4552# 4553# 4555# 4556# 4559# 4561# 4574# 4575# 4576# 4577# 4579# 4580# 4583# 4586# 4605#
4606# 4613# 4626# 4627# 4628# 4629# 4631# 4632# 4635# 4637# 4650# 4651# 4652# 4653# 4655#
4656# 4659# 4661# 4674# 4675# 4676# 4677# 4679# 4680# 4683# 4686# 4705# 4706# 4713# 4726#
4727# 4728# 4729# 4731# 4732# 4735# 4737# 4750# 4751# 4752# 4753# 4755# 4756# 4759# 4761#
4774# 4775# 4776# 4777# 4779# 4780# 4783# 4786# 4805# 4806# 4812# 4826# 4827# 4828# 4829#
4831# 4832# 4835# 4837# 4852# 4853# 4854# 4855# 4857# 4858# 4861# 4863# 4876# 4877# 4878#
4879# 4881# 4882# 4885# 4888# 4907# 4908# 4914# 4928# 4929# 4930# 4931# 4933# 49[illegible]# 4937#

	4939#	4954#	4955#	4956#	4957#	4959#	4960#	4963#	4965#	4978#	4979#	4980#	4981#	4983#	4984#
	4987#	4990#	5009#	5010#	5016#	5029#	5030#	5031#	5032#	5034#	5035#	5038#	5040#	5055#	5056#
	5057#	5058#	5060#	5061#	5064#	5066#	5079#	5080#	5081#	5082#	5084#	5085#	5088#	5091#	5110#
	5111#	5117#	5130#	5131#	5132#	5133#	5135#	5136#	5139#	5141#	5156#	5157#	5158#	5159#	5161#
	5162#	5165#	5167#	5180#	5181#	5182#	5183#	5185#	5186#	5189#	5192#	5212#	5213#	5219#	5232#
	5233#	5234#	5235#	5237#	5238#	5241#	5243#	5258#	5259#	5260#	5261#	5263#	5264#	5267#	5269#
	5282#	5283#	5284#	5285#	5287#	5288#	5291#	5294#	5313#	5314#	5320#	5333#	5334#	5335#	5336#
	5338#	5339#	5342#	5344#	5359#	5360#	5361#	5362#	5364#	5365#	5368#	5370#	5383#	5384#	5385#
	5386#	5388#	5389#	5392#	5395#	5412#	5413#	5456#	5457#	5458#	5459#	5465#	5466#	5473#	5493#
	5494#	5527#	5528#	5529#	5530#	5535#	5536#	5545#	5588#	5589#	5645#	5646#	5647#	5648#	5680#
	5681#	5728#	5729#	5730#	5731#	5736#	5776#	5777#	5833#	5834#	5835#	5836#	5868#	5869#	5878#
	5893#	5894#	5908#	5909#	5910#	5911#	5914#	5915#	5927#	5928#	5929#	5930#	5933#	5934#	5944#
	5945#	5946#	5947#	5951#	5967#	5968#	5982#	5983#	5984#	5985#	5988#	5989#	6000#	6001#	6002#
	6003#	6006#	6007#	6017#	6018#	6019#	6020#	6024#	6054#	6055#	6056#	6057#	6071#	6072#	6073#
	6074#	6087#	6088#	6089#	6090#	6097#	6113#	6130#	6131#	6132#	6133#	6152#	6153#	6154#	6155#
	6165#	6166#	6167#	6168#	6173#	6174#	6182#	6183#	6184#	6185#	6195#	6196#	6197#	6198#	6207#
	6333#	6334#	6335#	6336#	6338#	6339#	6364#	6365#	6366#	6367#	6386#	6387#	6388#	6389#	6391#
	6392#	6412#	6413#	6414#	6415#	6417#	6418#	6423#	6424#	6436#	6484#	6485#	6486#	6487#	6492#
	6508#	6509#	6539#	6540#	6541#	6542#	6546#	6547#	6552#	6553#	6561#	6579#	6580#	6629#	6630#
	6631#	6632#	6635#	6636#	6644#	6645#	6646#	6647#	6651#	6652#	6659#	6660#	6661#	6662#	6665#
	6666#	6673#	6674#	6675#	6676#	6679#	6680#	6689#	6690#	6691#	6692#	6695#	6696#	6701#	6702#
	6703#	6704#	6706#	6707#	6713#	6714#	6715#	6716#	6718#	6719#	6725#	6726#	6727#	6728#	6731#
	6732#	6753#	6788#	6789#	6807#	6808#	6809#	6810#	6815#	6838#	6839#	6840#	6841#	6843#	6844#
	6852#	6853#	6854#	6855#	6861#	6878#	6879#	6908#	6909#	6910#	6911#	6915#	6916#	6924#	6944#
	6945#	6966#	6967#	6968#	6969#	6975#	7013#	7014#	7015#	7016#	7021#	7041#	7042#	7079#	7080#
	7081#	7082#	7088#	7108#	7109#	7132#	7134#	7135#	7136#	7137#	7140#	7158#	7159#	7160#	7161#
	7164#	7181#	7182#	7183#	7184#	7188#	7191#	7204#	7205#	7206#	7207#	7215#	7230#	7234#	7235
	7236	7237	7238	7240#	7241	7242	7243	7251#	7252	7253	7254	7255	7257#	7342#	7347#
	7358#	7359#	7360#												
M$GNLS	1#	1250#	3412#	3463#	3519#	3542#	4033#	4057#	4081#	4132#	4155#	4178#	4231#	4256#	4280#
	4332#	4356#	4380#	4433#	4457#	4481#	4534#	4558#	4582#	4634#	4658#	4682#	4734#	4758#	4782#
	4834#	4860#	4884#	4936#	4962#	4986#	5037#	5063#	5087#	5138#	5164#	5188#	5240#	5266#	5290#
	5341#	5367#	5391#												
M$GNSU	1#	1250#													
M$GNTA	1#	1250#	1463#	1478#	2389#	2418#	2447#	2475#	2503#	2532#	2561#	2589#	2617#	2645#	2673#
	2701#	2730#	2758#	2781#	2809#	2837#	2849#	2868#	2885#	2902#	2922#	2941#	2966#	2985#	3005#
	3026#	3177#	3203#	3219#	3239#	3257#	3309#	3372#	3422#	3474#	3550#	3612#	3674#	3765#	3894#
	3981#	4084#	4181#	4283#	4383#	4484#	4585#	4685#	4785#	4887#	4989#	5090#	5191#	5293#	5394#
	5472#	5544#	5735#	5877#	5950#	6023#	6096#	6206#	6435#	6491#	6560#	6752#	6814#	6860#	6923#
	6974#	7020#	7087#	7214#	7257#	7258	7347#	7348							
M$GNTE	1#	1250#	3280#	3323#	3383#	3433#	3486#	3562#	3623#	3686#	3777#	3907#	3998#	4101#	4198#
	4300#	4400#	4501#	4602#	4702#	4802#	4904#	5006#	5107#	5209#	5310#	5409#	5490#	5583#	5771#
	5890#	5964#	6035#	6111#	6221#	6449#	6504#	6574#	6767#	6828#	6874#	6940#	6991#	7037#	7103#
M$HAPT	1#	1250#	1282#												
M$HNAP	1#	1250#	1282#	1321											
M$INCR	1#	1250#	1256#	1370#	1445#	1472#	2363#	2368#	2374#	2382#	2387#	2390#	2392#	2397#	2403#
	2411#	2416#	2419#	2421#	2426#	2432#	2440#	2445#	2448#	2450#	2455#	2461#	2468#	2473#	2476#
	2478#	2483#	2489#	2496#	2501#	2504#	2506#	2511#	2517#	2525#	2530#	2533#	2535#	2540#	2546#
	2554#	2559#	2562#	2564#	2569#	2575#	2582#	2587#	2590#	2592#	2597#	2603#	2610#	2615#	2618#
	2620#	2625#	2631#	2638#	2643#	2646#	2648#	2653#	2659#	2666#	2671#	2674#	2676#	2681#	2687#
	2694#	2699#	2702#	2704#	2709#	2715#	2723#	2728#	2731#	2733#	2738#	2744#	2751#	2756#	2759#
	2763#	2768#	2774#	2779#	2782#	2784#	2789#	2795#	2802#	2807#	2810#	2812#	2817#	2823#	2830#
	2835#	2838#	2841#	2846#	2850#	2853#	2859#	2865#	2869#	2871#	2876#	2882#	2886#	2888#	2893#
	2899#	2903#	2906#	2913#	2919#	2923#	2925#	2932#	2938#	2942#	2944#	2951#	2957#	2963#	2967#
	2969#	2976#	2982#	2986#	2989#	2996#	3002#	3006#	3017#	3027#	3041#	3058#	3064#	3070#	3076#
	3093#	3178#	3182#	3198#	3204#	3214#	3216#	3220#	3234#	3237#	3240#	3255#	3258#	3280#	3281#

```
            3288#  3297#  3305#  3310#  3323#  3324#  3345#  3351#  3365#  3373#  3383#  3384#  3396#  3395#  3404#
            3409#  3413#  3423#  3433#  3434#  3436#  3445#  3455#  3460#  3464#  3475#  3486#  3487#  3489#  3497#
            3511#  3516#  3520#  3522#  3534#  3539#  3543#  3551#  3562#  3563#  3594#  3599#  3606#  3613#  3623#
            3624#  3656#  3661#  3668#  3675#  3686#  3687#  3718#  3723#  3751#  3756#  3759#  3766#  3777#  3778#
            3780#  3816#  3841#  3849#  3852#  3888#  3895#  3907#  3908#  3910#  3933#  3938#  3956#  3975#  3982#
            3998#  3999#  4002#  4010#  4025#  4030#  4034#  4036#  4049#  4054#  4058#  4060#  4073#  4078#  4082#
            4085#  4101#  4102#  4104#  4112#  4124#  4129#  4133#  4135#  4147#  4152#  4156#  4158#  4170#  4175#
            4179#  4182#  4198#  4199#  4201#  4209#  4222#  4228#  4232#  4234#  4247#  4253#  4257#  4259#  4272#
            4277#  4281#  4284#  4300#  4301#  4303#  4311#  4324#  4329#  4333#  4335#  4348#  4353#  4357#  4359#
            4372#  4377#  4381#  4384#  4400#  4401#  4403#  4412#  4425#  4430#  4434#  4436#  4449#  4454#  4458#
            4460#  4473#  4478#  4482#  4485#  4501#  4502#  4504#  4513#  4526#  4531#  4535#  4537#  4550#  4555#
            4559#  4561#  4574#  4579#  4583#  4586#  4602#  4603#  4605#  4613#  4626#  4631#  4635#  4637#  4650#
            4655#  4659#  4661#  4674#  4679#  4683#  4686#  4702#  4703#  4705#  4713#  4726#  4731#  4735#  4737#
            4750#  4755#  4759#  4761#  4774#  4779#  4783#  4786#  4802#  4803#  4805#  4812#  4826#  4831#  4835#
            4837#  4852#  4857#  4861#  4863#  4876#  4881#  4885#  4888#  4904#  4905#  4907#  4914#  4928#  4933#
            4937#  4939#  4954#  4959#  4963#  4965#  4978#  4983#  4987#  4990#  5006#  5007#  5009#  5016#  5029#
            5034#  5038#  5040#  5055#  5060#  5064#  5066#  5079#  5084#  5088#  5091#  5107#  5108#  5110#  5117#
            5130#  5135#  5139#  5141#  5156#  5161#  5165#  5167#  5180#  5185#  5189#  5192#  5209#  5210#  5212#
            5219#  5232#  5237#  5241#  5243#  5258#  5263#  5267#  5269#  5282#  5287#  5291#  5294#  5310#  5311#
            5313#  5320#  5333#  5338#  5342#  5344#  5359#  5364#  5368#  5370#  5383#  5388#  5392#  5395#  5409#
            5410#  5412#  5456#  5465#  5473#  5490#  5491#  5493#  5527#  5535#  5545#  5583#  5584#  5588#  5645#
            5680#  5728#  5736#  5771#  5772#  5776#  5833#  5868#  5878#  5890#  5891#  5893#  5908#  5914#  5927#
            5933#  5944#  5951#  5964#  5965#  5967#  5982#  5988#  6000#  6006#  6017#  6024#  6035#  6036#  6054#
            6071#  6087#  6097#  6111#  6112#  6113#  6130#  6152#  6165#  6173#  6182#  6195#  6207#  6221#  6222#
            6333#  6338#  6364#  6386#  6391#  6412#  6417#  6423#  6436#  6449#  6450#  6484#  6492#  6504#  6505#
            6508#  6539#  6546#  6552#  6561#  6574#  6575#  6579#  6629#  6635#  6644#  6651#  6659#  6665#  6673#
            6679#  6689#  6695#  6701#  6706#  6713#  6718#  6725#  6731#  6753#  6767#  6768#  6788#  6807#  6815#
            6828#  6829#  6838#  6843#  6852#  6861#  6874#  6875#  6878#  6908#  6915#  6924#  6940#  6941#  6944#
            6966#  6975#  6991#  6992#  7013#  7021#  7037#  7038#  7041#  7079#  7088#  7103#  7104#  7109#  7132#
            7134#  7140#  7158#  7164#  7181#  7188#  7191#  7204#  7215#  7230#  7342#
M$IOSE        1#  1250#
M$LDRO        1#  1250#  3057#  3063#  3069#  3075#  3092#  3197#  7108#
M$MASK        1#  1250#
M$MCHI        1#  1250#
M$MCLO        1#  1250#
M$MSK1        1#  1250#
M$POP         1#  1250#  1375#  1463#  1478#  2389#  2418#  2447#  2475#  2503#  2532#  2561#  2589#  2617#  2645#
            2673#  2701#  2730#  2758#  2781#  2809#  2837#  2849#  2868#  2885#  2902#  2922#  2941#  2966#  2985#
            3005#  3026#  3177#  3203#  3219#  3239#  3257#  3309#  3372#  3412#  3422#  3463#  3474#  3519#  3542#
            3550#  3612#  3674#  3765#  3894#  3981#  4033#  4057#  4081#  4084#  4132#  4155#  4178#  4181#  4231#
            4256#  4280#  4283#  4332#  4356#  4380#  4383#  4433#  4457#  4481#  4484#  4534#  4558#  4582#  4585#
            4634#  4658#  4682#  4685#  4734#  4758#  4782#  4785#  4834#  4860#  4884#  4887#  4936#  4962#  4986#
            4989#  5037#  5063#  5087#  5090#  5138#  5164#  5188#  5191#  5240#  5266#  5290#  5293#  5341#  5367#
            5391#  5394#  5472#  5544#  5735#  5877#  5950#  6023#  6096#  6206#  6435#  6491#  6560#  6752#  6814#
            6860#  6923#  6974#  7020#  7087#  7214#  7257#  7347#
M$PRIN        1#  1250#  2364#  2370#  2376#  2384#  2393#  2399#  2405#  2413#  2422#  2428#  2434#  2442#  2451#
            2457#  2463#  2470#  2479#  2485#  2491#  2498#  2507#  2513#  2519#  2527#  2536#  2542#  2548#  2556#
            2565#  2571#  2577#  2584#  2593#  2599#  2605#  2612#  2621#  2627#  2633#  2640#  2649#  2655#  2661#
            2668#  2677#  2683#  2689#  2696#  2705#  2711#  2717#  2725#  2734#  2740#  2746#  2753#  2764#  2770#
            2776#  2785#  2791#  2797#  2804#  2813#  2819#  2825#  2832#  2843#  2855#  2862#  2873#  2879#  2890#
            2896#  2908#  2916#  2927#  2935#  2946#  2954#  2960#  2971#  2979#  2991#  2999#
M$PUSH        1#  1250#  1256#  1370#  1445#  1472#  2363#  2392#  2421#  2450#  2478#  2506#  2535#  2564#  2592#
            2620#  2648#  2676#  2704#  2733#  2763#  2784#  2812#  2841#  2853#  2871#  2888#  2906#  2925#  2944#
            2969#  2989#  3017#  3041#  3182#  3214#  3234#  3255#  3280#  3281    3323#  3324    3383#  3384    3395#
            3433#  3434    3445#  3486#  3487    3497#  3522#  3562#  3563    3623#  3624    3686#  3687    3777#  3778
            3907#  3908    3998#  3999    4010#  4036#  4060#  4101#  4102    4112#  4135#  4158#  4198#  4199    4209#
```

```
                4234#  4259#  4300#  4301   4311#  4335#  4359#  4400#  4401   4412#  4436#  4460#  4501#  4502   4513#
                4537#  4561#  4602#  4603   4613#  4637#  4661#  4702#  4703   4713#  4737#  4761#  4802#  4803   4812#
                4837#  4863#  4904#  4905   4914#  4939#  4965#  5006#  5007   5016#  5040#  5066#  5107#  5108   5117#
                5141#  5167#  5209#  5210   5219#  5243#  5269#  5310#  5311   5320#  5344#  5370#  5409#  5410   5490#
                5491   5583#  5584   5771#  5772   5890#  5891   5964#  5965   6035#  6036   6111#  6112   6221#  6222
                6449#  6450   6504#  6505   6574#  6575   6767#  6768   6828#  6829   6874#  6875   6940#  6941   6991#
                6992   7037#  7038   7103#  7104   7230#  7342#
M$PUT             1#   1250#  2364#  2370#  2376#  2384#  2393#  2399#  2405#  2413#  2422#  2428#  2434#  2442#  2451#
                2457#  2463#  2470#  2479#  2485#  2491#  2498#  2507#  2513#  2519#  2527#  2536#  2542#  2548#  2556#
                2565#  2571#  2577#  2584#  2593#  2599#  2605#  2612#  2621#  2627#  2633#  2640#  2649#  2655#  2661#
                2668#  2677#  2683#  2689#  2696#  2705#  2711#  2717#  2725#  2734#  2740#  2746#  2753#  2764#  2770#
                2776#  2785#  2791#  2797#  2804#  2813#  2819#  2825#  2832#  2843#  2855#  2862#  2873#  2879#  2890#
                2896#  2908#  2916#  2927#  2935#  2946#  2954#  2960#  2971#  2979#  2991#  2999#
M$PUT1            1#   1250#  2364#  2365   2366   2370#  2371   2372   2376#  2377   2378   2379   2380   2384#  2385
                2393#  2394   2395   2399#  2400   2401   2405#  2406   2407   2408   2409   2413#  2414   2422#  2423
                2424   2428#  2429   2430   2434#  2435   2436   2437   2438   2442#  2443   2451#  2452   2453   2457#
                2458   2459   2463#  2464   2465   2466   2470#  2471   2479#  2480   2481   2485#  2486   2487   2491#
                2492   2493   2494   2498#  2499   2507#  2508   2509   2513#  2514   2515   2519#  2520   2521   2522
                2523   2527#  2528   2536#  2537   2538   2542#  2543   2544   2548#  2549   2550   2551   2552   2556#
                2557   2565#  2566   2567   2571#  2572   2573   2577#  2578   2579   2580   2584#  2585   2593#  2594
                2595   2599#  2600   2601   2605#  2606   2607   2608   2612#  2613   2621#  2622   2623   2627#  2628
                2629   2633#  2634   2635   2636   2640#  2641   2649#  2650   2651   2655#  2656   2657   2661#  2662
                2663   2664   2668#  2669   2677#  2678   2679   2683#  2684   2685   2689#  2690   2691   2692   2696#
                2697   2705#  2706   2707   2711#  2712   2713   2717#  2718   2719   2720   2721   2725#  2726   2734#
                2735   2736   2740#  2741   2742   2746#  2747   2748   2749   2753#  2754   2764#  2765   2766   2770#
                2771   2772   2776#  2777   2785#  2786   2787   2791#  2792   2793   2797#  2798   2799   2800   2804#
                2805   2813#  2814   2815   2819#  2820   2821   2825#  2826   2827   2828   2832#  2833   2843#  2844
                2855#  2856   2857   2862#  2863   2873#  2874   2879#  2880   2890#  2891   2896#  2897   2908#  2909
                2910   2911   2916#  2917   2927#  2928   2929   2930   2935#  2936   2946#  2947   2948   2949   2954#
                2955   2960#  2961   2971#  2972   2973   2974   2979#  2980   2991#  2992   2993   2994   2999#  3000
M$RADI            1#   1250#  7234#  7240#  7251#
M$RBRO            1#   1250#
M$RNRO            1#   1250#  3092#  3094
M$SETS            1#   1250#  1256#  1370#  1445#  1472#  2363#  2392#  2421#  2450#  2478#  2506#  2535#  2564#  2592#
                2620#  2648#  2676#  2704#  2733#  2763#  2784#  2812#  2841#  2853#  2871#  2888#  2906#  2925#  2944#
                2969#  2989#  3017#  3041#  3182#  3214#  3234#  3255#  3281#  3324#  3384#  3395#  3434#  3445#  3487#
                3497#  3522#  3563#  3624#  3687#  3778#  3908#  3999#  4010#  4036#  4060#  4102#  4112#  4135#  4158#
                4199#  4209#  4234#  4259#  4301#  4311#  4335#  4359#  4401#  4412#  4436#  4460#  4502#  4513#  4537#
                4561#  4603#  4613#  4637#  4661#  4703#  4713#  4737#  4761#  4803#  4812#  4837#  4863#  4905#  4914#
                4939#  4965#  5007#  5016#  5040#  5066#  5108#  5117#  5141#  5167#  5210#  5219#  5243#  5269#  5311#
                5320#  5344#  5370#  5410#  5491#  5584#  5772#  5891#  5965#  6036#  6112#  6222#  6450#  6505#  6575#
                6768#  6829#  6875#  6941#  6992#  7038#  7104#  7230#  7342#
M$STAR            1#   1250#
M$SVC             1#   1250#  2364#  2368   2370#  2374   2376#  2382   2384#  2387   2389#  2390   2393#  2397   2399#
                2403   2405#  2411   2413#  2416   2418#  2419   2422#  2426   2428#  2432   2434#  2440   2442#  2445
                2447#  2448   2451#  2455   2457#  2461   2463#  2468   2470#  2473   2475#  2476   2479#  2483   2485#
                2489   2491#  2496   2498#  2501   2503#  2504   2507#  2511   2513#  2517   2519#  2525   2527#  2530
                2532#  2533   2536#  2540   2542#  2546   2548#  2554   2556#  2559   2561#  2562   2565#  2569   2571#
                2575   2577#  2582   2584#  2587   2589#  2590   2593#  2597   2599#  2603   2605#  2610   2612#  2615
                2617#  2618   2621#  2625   2627#  2631   2633#  2638   2640#  2643   2645#  2646   2649#  2653   2655#
                2659   2661#  2666   2668#  2671   2673#  2674   2677#  2681   2683#  2687   2689#  2694   2696#  2699
                2701#  2702   2705#  2709   2711#  2715   2717#  2723   2725#  2728   2730#  2731   2734#  2738   2740#
                2744   2746#  2751   2753#  2756   2758#  2759   2764#  2768   2770#  2774   2776#  2779   2781#  2782
                2785#  2789   2791#  2795   2797#  2802   2804#  2807   2809#  2810   2813#  2817   2819#  2823   2825#
                2830   2832#  2835   2837#  2838   2843#  2846   2849#  2850   2855#  2859   2862#  2865   2868#  2869
                2873#  2876   2879#  2882   2885#  2886   2890#  2893   2896#  2899   2902#  2903   2908#  2913   2916#
```

2919 2922# 2923 2927# 2932 2935# 2938 2941# 2942 2946# 2951 2954# 2957 2960# 2963
2966# 2967 2971# 2976 2979# 2982 2985# 2986 2991# 2996 2999# 3002 3005# 3006 3021#
3026# 3027 3057# 3058 3063# 3064 3069# 3070 3075# 3076 3092# 3093 3177# 3178 3197#
3198 3203# 3204 3216# 3219# 3220 3237# 3239# 3240 3257# 3258 3288# 3297 3305# 3309#
3310 3345 3351# 3365 3372# 3373 3386# 3395# 3404 3409# 3412# 3413 3422# 3423 3436#
3445# 3455 3460# 3463# 3464 3474# 3475 3489# 3497# 3511 3516# 3519# 3520 3522# 3534
3539# 3542# 3543 3550# 3551 3594 3599# 3606# 3612# 3613 3656 3661# 3668# 3674# 3675
3718 3723# 3751 3756# 3759# 3765# 3766 3780# 3816 3841 3849# 3852# 3888 3894# 3895
3910# 3933 3938# 3956 3975 3981# 3982 4002# 4010# 4025 4030# 4033# 4034 4036# 4049
4054# 4057# 4058 4060# 4073 4078# 4081# 4082 4084# 4085 4104# 4112# 4124 4129# 4132#
4133 4135# 4147 4152# 4155# 4156 4158# 4170 4175# 4178# 4179 4181# 4182 4201# 4209#
4222 4228# 4231# 4232 4234# 4247 4253# 4256# 4257 4259# 4272 4277# 4280# 4281 4283#
4284 4303# 4311# 4324 4329# 4332# 4333 4335# 4348 4353# 4356# 4357 4359# 4372 4377#
4380# 4381 4383# 4384 4403# 4412# 4425 4430# 4433# 4434 4436# 4449 4454# 4457# 4458
4460# 4473 4478# 4481# 4482 4484# 4485 4504# 4513# 4526 4531# 4534# 4535 4537# 4550
4555# 4558# 4559 4561# 4574 4579# 4582# 4583 4585# 4586 4605# 4613# 4626 4631# 4634#
4635 4637# 4650 4655# 4658# 4659 4661# 4674 4679# 4682# 4683 4685# 4686 4705# 4713#
4726 4731# 4734# 4735 4737# 4750 4755# 4758# 4759 4761# 4774 4779# 4782# 4783 4785#
4786 4805# 4812# 4826 4831# 4834# 4835 4837# 4852 4857# 4860# 4861 4863# 4876 4881#
4884# 4885 4887# 4888 4907# 4914# 4928 4933# 4936# 4937 4939# 4954 4959# 4962# 4963
4965# 4978 4983# 4986# 4987 4989# 4990 5009# 5016# 5029 5034# 5037# 5038 5040# 5055
5060# 5063# 5064 5066# 5079 5084# 5087# 5088 5090# 5091 5110# 5117# 5130 5135# 5138#
5139 5141# 5156 5161# 5164# 5165 5167# 5180 5185# 5188# 5189 5191# 5192 5212# 5219#
5232 5237# 5240# 5241 5243# 5258 5263# 5266# 5267 5269# 5282 5287# 5290# 5291 5293#
5294 5313# 5320# 5333 5338# 5341# 5342 5344# 5359 5364# 5367# 5368 5370# 5383 5388#
5391# 5392 5394# 5395 5412# 5456 5465# 5472# 5473 5493# 5527 5535# 5544# 5545 5588#
5645 5680# 5728 5735# 5736 5776# 5833 5868# 5877# 5878 5893# 5908 5914# 5927 5933#
5944 5950# 5951 5967# 5982 5988# 6000 6006# 6017 6023# 6024 6054 6071 6087 6096#
6097 6113# 6130 6152 6165 6173# 6182 6195 6206# 6207 6333 6338# 6364 6386 6391#
6412 6417# 6423# 6435# 6436 6484 6491# 6492 6508# 6539 6546# 6552# 6560# 6561 6579#
6629 6635# 6644 6651# 6659 6665# 6673 6679# 6689 6695# 6701 6706# 6713 6718# 6725
6731# 6752# 6753 6788# 6807 6814# 6815 6838 6843# 6852 6860# 6861 6878# 6908 6915#
6923# 6924 6944# 6966 6974# 6975 7013 7020# 7021 7041# 7079 7087# 7088 7108# 7109
7132# 7134 7140# 7158 7164# 7181 7188# 7191# 7204 7214# 7215

M$TLAB 1# 1250# 2368# 2374# 2382# 2387# 2390# 2397# 2403# 2411# 2416# 2419# 2426# 2432# 2440#
2445# 2448# 2455# 2461# 2468# 2473# 2476# 2483# 2489# 2496# 2501# 2504# 2511# 2517# 2525#
2530# 2533# 2540# 2546# 2554# 2559# 2562# 2569# 2575# 2582# 2587# 2590# 2597# 2603# 2610#
2615# 2618# 2625# 2631# 2638# 2643# 2646# 2653# 2659# 2666# 2671# 2674# 2681# 2687# 2694#
2699# 2702# 2709# 2715# 2723# 2728# 2731# 2738# 2744# 2751# 2756# 2759# 2768# 2774# 2779#
2782# 2789# 2795# 2802# 2807# 2810# 2817# 2823# 2830# 2835# 2838# 2846# 2850# 2859# 2865#
2869# 2876# [illegible] 2886# 2893# 2899# 2903# 2913# 2919# 2923# 2932# 2938# 2942# 2951# 2957#
2963# 2967# 2976# 2982# 2986# 2996# 3002# 3006# 3027# 3058# 3064# 3070# 3076# 3093# 3178#
3198# 3204# 3216# 3220# 3237# 3240# 3258# 3288# 3297# 3305# 3310# 3345# 3351# 3365# 3373#
3386# 3395# 3404# 3409# 3413# 3423# 3436# 3445# 3455# 3460# 3464# 3475# 3489# 3497# 3511#
3516# 3520# 3522# 3534# 3539# 3543# 3551# 3594# 3599# 3606# 3613# 3656# 3661# 3668# 3675#
3718# 3723# 3751# 3756# 3759# 3766# 3780# 3816# 3841# 3849# 3852# 3888# 3895# 3910# 3933#
3938# 3956# 3975# 3982# 4002# 4010# 4025# 4030# 4034# 4036# 4049# 4054# 4058# 4060# 4073#
4078# [illegible] 4085# 4104# 4112# 4124# 4129# 4133# 4135# 4147# 4152# 4156# 4158# 4170# 4175#
4179# 4182# 4201# 4209# 4222# 4228# 4232# 4234# 4247# 4253# 4257# 4259# 4272# 4277# 4281#
4284# 4303# 4311# 4324# 4329# 4333# 4335# 4348# 4353# 4357# 4359# 4372# 4377# 4381# 4384#
4403# 4412# 4425# 4430# 4434# 4436# 4449# 4454# 4458# 4460# 4473# 4478# 4482# 4485# 4504#
4513# 4526# 4531# 4535# 4537# 4550# 4555# 4559# 4561# 4574# 4579# 4583# 4586# 4605# 4613#
4626# 4631# 4635# 4637# 4650# 4655# 4659# 4661# 4674# 4679# 4683# 4686# 4705# 4713# 4726#
4731# 4735# 4737# 4750# 4755# 4759# 4761# 4774# 4779# 4783# 4786# 4805# 4812# 4826# 4831#
4835# 4837# 4852# 4857# 4861# 4863# 4876# 4881# 4885# 4888# 4907# 4914# 4928# 4933# 4937#
4939# 4954# 4959# 4963# 4965# 4978# 4983# 4987# 4990# 5009# 5016# 5029# 5034# 5038# 5040#

```
                5055#   5060#   5064#   5066#   5079#   5084#   5088#   5091#   5110#   5117#   5130#   5135#   5139#   5141#   5156#
                5161#   5165#   5167#   5180#   5185#   5189#   5192#   5212#   5219#   5232#   5237#   5241#   5243#   5258#   5263#
                5267#   5269#   5282#   5287#   5291#   5294#   5313#   5320#   5333#   5338#   5342#   5344#   5359#   5364#   5368#
                5370#   5383#   5388#   5392#   5395#   5412#   5456#   5465#   5473#   5493#   5527#   5535#   5545#   5588#   5645#
                5680#   5728#   5736#   5776#   5833#   5868#   5878#   5893#   5908#   5914#   5927#   5933#   5944#   5951#   5967#
                5932#   5988#   6000#   6006#   6017#   6024#   6054#   6071#   6087#   6097#   6113#   6130#   6152#   6165#   6173#
                6182#   6195#   6207#   6333#   6338#   6364#   6386#   6391#   6412#   6417#   6423#   6436#   6484#   6492#   6508#
                6539#   6546#   6552#   6561#   6579#   6629#   6635#   6644#   6651#   6659#   6665#   6673#   6679#   6689#   6695#
                6701#   6706#   6713#   6718#   6725#   6731#   6753#   6788#   6807#   6815#   6838#   6843#   6852#   6861#   6878#
                6908#   6915#   6924#   6944#   6966#   6975#   7013#   7021#   7041#   7079#   7088#   7109#   7132#   7134#   7140#
                7158#   7164#   7181#   7188#   7191#   7204#   7215#
M$TSTL             1#   1250#   2368#   2374#   2382#   2387#   2390#   2397#   2403#   2411#   2416#   2419#   2426#   2432#   2440#
                2445#   2448#   2455#   2461#   2468#   2473#   2476#   2483#   2439#   2496#   2501#   2504#   2511#   2517#   2525#
                2530#   2533#   2540#   2546#   2554#   2559#   2562#   2569#   2575#   2582#   2587#   2590#   2597#   2603#   2610#
                2615#   2618#   2625#   2631#   2638#   2643#   2646#   2653#   2659#   2666#   2671#   2674#   2681#   2687#   2694#
                2699#   2702#   2709#   2715#   2723#   2728#   2731#   2738#   2744#   2751#   2756#   2759#   2768#   2774#   2779#
                2782#   2789#   2795#   2802#   2807#   2810#   2817#   2823#   2830#   2835#   2838#   2846#   2850#   2859#   2865#
                2869#   2876#   2882#   2886#   2893#   2899#   2903#   2913#   2919#   2923#   2932#   2938#   2942#   2951#   2957#
                2963#   2967#   2976#   2982#   2986#   2996#   3002#   3006#   3027#   3058#   3064#   3070#   3076#   3093#   3178#
                3198#   3204#   3216#   3220#   3237#   3240#   3258#   3288#   3297#   3305#   3310#   3345#   3351#   3365#   3373#
                3386#   3395#   3404#   3409#   3413#   3423#   3436#   3445#   3455#   3460#   3464#   3475#   3489#   3497#   3511#
                3516#   3520#   3522#   3534#   3539#   3543#   3551#   3594#   3599#   3606#   3613#   3656#   3661#   3668#   3675#
                3718#   3723#   3751#   3756#   3759#   3766#   3780#   3816#   3841#   3849#   3852#   3888#   3895#   3910#   3933#
                3938#   3956#   3975#   3982#   4002#   4010#   4025#   4030#   4034#   4036#   4049#   4054#   4058#   4060#   4073#
                4078#   4082#   4085#   4104#   4112#   4124#   4129#   4133#   4135#   4147#   4152#   4156#   4158#   4170#   4175#
                4179#   4182#   4201#   4209#   4222#   4228#   4232#   4234#   4247#   4253#   4257#   4259#   4272#   4277#   4281#
                4284#   4303#   4311#   4324#   4329#   4333#   4335#   4348#   4353#   4357#   4359#   4372#   4377#   4381#   4384#
                4403#   4412#   4425#   4430#   4434#   4436#   4449#   4454#   4458#   4460#   4473#   4478#   4482#   4485#   4504#
                4513#   4526#   4531#   4535#   4537#   4550#   4555#   4559#   4561#   4574#   4579#   4583#   4586#   4605#   4613#
                4626#   4631#   4635#   4637#   4650#   4655#   4659#   4661#   4674#   4679#   4683#   4686#   4705#   4713#   4726#
                4731#   4735#   4737#   4750#   4755#   4759#   4761#   4774#   4779#   4783#   4786#   4805#   4812#   4826#   4831#
                4835#   4837#   4852#   4857#   4861#   4863#   4876#   4881#   4885#   4888#   4907#   4914#   4928#   4933#   4937#
                4939#   4954#   4959#   4963#   4965#   4978#   4983#   4987#   4990#   5009#   5016#   5029#   5034#   5038#   5040#
                5055#   5060#   5064#   5066#   5079#   5084#   5088#   5091#   5110#   5117#   5130#   5135#   5139#   5141#   5156#
                5161#   5165#   5167#   5180#   5185#   5189#   5192#   5212#   5219#   5232#   5237#   5241#   5243#   5258#   5263#
                5267#   5269#   5282#   5287#   5291#   5294#   5313#   5320#   5333#   5338#   5342#   5344#   5359#   5364#   5368#
                5370#   5383#   5388#   5392#   5395#   5412#   5456#   5465#   5473#   5493#   5527#   5535#   5545#   5588#   5645#
                5680#   5728#   5736#   5776#   5833#   5868#   5878#   5893#   5908#   5914#   5927#   5933#   5944#   5951#   5967#
                5982#   5988#   6000#   6006#   6017#   6024#   6054#   6071#   6087#   6097#   6113#   6130#   6152#   6165#   6173#
                6182#   6195#   6207#   6333#   6338#   6364#   6386#   6391#   6412#   6417#   6423#   6436#   6484#   6492#   6508#
                6539#   6546#   6552#   6561#   6579#   6629#   6635#   6644#   6651#   6659#   6665#   6673#   6679#   6689#   6695#
                6701#   6706#   6713#   6718#   6725#   6731#   6753#   6788#   6807#   6815#   6838#   6843#   6852#   6861#   6878#
                6908#   6915#   6924#   6944#   6966#   6975#   7013#   7021#   7041#   7079#   7088#   7109#   7132#   7134#   7140#
                7158#   7164#   7181#   7188#   7191#   7204#   7215#
M$WORD             1#   1250#   1321#   1330    1384#   1386    1387    1388    1389    1390    1391    1392    1393    1394    1395
                1396    1397    1398    1399    1400    1401    1402    1403    1404    1405    1406    1407    1408    1409    1410
                1411    1412    1413    1414    1415    1416    1417    1418    1419    1420    1421    1422    1423    1424    1425
                1426    1427    1428    3021#   3297#   3298    3299    3300    3345#   3346    3347    3348    3365#   3366    3367
                3368    3386#   3404#   3405    3406    3407    3436#   3455#   3456    3457    3458    3489#   3511#   3512    3513
                3514    3534#   3535    3536    3537    3594#   3595    3596    3597    3656#   3657    3658    3659    3718#   3719
                3720    3721    3751#   3752    3753    3754    3780#   3816#   3817    3818    3819    3841#   3842    3843    3844
                3888#   3889    3890    3891    3910#   3933#   3934    3935    3936    3956#   3957    3958    3959    3975#   3976
                3977    3978    4002#   4025#   4026    4027    4028    4049#   4050    4051    4052    4073#   4074    4075    4076
                4104#   4124#   4125    4126    4127    4147#   4148    4149    4150    4170#   4171    4172    4173    4201#   4222#
                4223    4224    4225    4247#   4248    4249    4250    4272#   4273    4274    4275    4303#   4324#   4325    4326
                4327    4348#   4349    4350    4351    4372#   4373    4374    4375    4403#   4425#   4426    4427    4428    4449#
```

```
                4450    4451    4452    4473#   4474    4475    4476    4504#   4526#   4527    4528    4529    4550#   4551    4552
                4553    4574#   4575    4576    4577    4605#   4626#   4627    4628    4629    4650#   4651    4652    4653    4674#
                4675    4676    4677    4705#   4726#   4727    4728    4729    4750#   4751    4752    4753    4774#   4775    4776
                4777    4805#   4826#   4827    4828    4829    4852#   4853    4854    4855    4876#   4877    4878    4879    4907#
                4928#   4929    4930    4931    4954#   4955    4956    4957    4978#   4979    4980    4981    5009#   5029#   5030
                5031    5032    5055#   5056    5057    5058    5079#   5080    5081    5082    5110#   5130#   5131    5132    5133
                5156#   5157    5158    5159    5180#   5181    5182    5183    5212#   5232#   5233    5234    5235    5258#   5259
                5260    5261    5282#   5283    5284    5285    5313#   5333#   5334    5335    5336    5359#   5360    5361    5362
                5383#   5384    5385    5386    5412#   5456#   5457    5458    5459    5493#   5527#   5528    5529    5530    5588#
                5645#   5646    5647    5648    5728#   5729    5730    5731    5776#   5833#   5834    5835    5836    5893#   5908#
                5909    5910    5911    5927#   5928    5929    5930    5944#   5945    5946    5947    5967#   5982#   5983    5984
                5985    6000#   6001    6002    6003    6017#   6018    6019    6020    6054#   6055    6056    6057    6071#   6072
                6073    6074    6087#   6088    6089    6090    6130#   6131    6132    6133    6152#   6153    6154    6155    6165#
                6166    6167    6168    6173#   6182#   6183    6184    6185    6195#   6196    6197    6198    6333#   6334    6335
                6336    6364#   6365    6366    6367    6386#   6387    6388    6389    6412#   6413    6414    6415    6423#   6484#
                6485    6486    6487    6508#   6539#   6540    6541    6542    6552#   6579#   6629#   6630    6631    6632    6644#
                6645    6646    6647    6659#   6660    6661    6662    6673#   6674    6675    6676    6689#   6690    6691    6692
                6701#   6702    6703    6704    6713#   6714    6715    6716    6725#   6726    6727    6728    6731#   6788#   6807#
                6808    6809    6810    6838#   6839    6840    6841    6852#   6853    6854    6855    6878#   6908#   6909    6910
                6911    6944#   6966#   6967    6968    6969    7013#   7014    7015    7016    7041#   7079#   7080    7081    7082
                7134#   7135    7136    7137    7158#   7159    7160    7161    7181#   7182    7183    7184    7204#   7205    7206
                7207    7234#   7240#   7251#   7359    7360
MSXFER             1#   1250#
OPEN               1#   1250#
POINTE             1#   1250#   1278
POPSP2          1756#   6146
PRINTB             1#   1250#   2364    2370    2376    2384    2393    2399    2405    2413    2422    2428    2434    2442    2451
                2457    2463    2470    2479    2485    2491    2498    2507    2513    2519    2527    2536    2542    2548    2556
                2565    2571    2577    2584    2593    2599    2605    2612    2621    2627    2633    2640    2649    2655    2661
                2668    2677    2683    2689    2696    2705    2711    2717    2725    2734    2740    2746    2753    2764    2770
                2776    2785    2791    2797    2804    2813    2819    2825    2832    2842    2861    2878    2895    2915    2934
                2953    2959    2978    2998
PRINTF             1#   1250#   2854    2872    2889    2907    2926    2945    2970    2990
PRINTS             1#   1250#
PRINTX             1#   1250#
READBU             1#   1250#
READEF             1#   1250#   3056    3062    3068    3074
RFLAGS             1#   1250#
ROMCLK          1829#   1904    1907    1910    1920    1928    1936    1944    1952    1955    1958    1969    1992    1998    3330
                3333    3336    3354    3357    3573    3576    3580    3583    3586    3635    3638    3642    3645    3648    3697
                3700    3704    3707    3710    3734    3737    3740    3743    3789    3793    3795    3802    3806    3824    3828
                3868    3871    3874    3877    3917    3921    3941    3947    3960    3963    3966    5420    5423    5428    5444
                5447    5506    5509    5518    5612    5615    5622    5625    5704    5707    5710    5714    5800    5803    5810
                5813    5899    5917    5936    5973    5991    6009    6059    6229    6243    6247    6251    6265    6269    6273
                6280    6283    6287    6291    6295    6304    6307    6323    6356    6374    6378    6397    6400    6404    6452
                6455    6458    6461    6466    6470    6884    6887    6895    6899    6951    6955    6999    7006    7054    7058
                7061    7071    7112    7116    7119    7123    7145    7149    7168    7171    7195
SETPRI             1#   1250#   7107
SETVEC             1#   1250#
SKIP04          1854#   3999
SKIP06          1840#   1984    3503    3860    4838    4940    5041    5142    5244    5345    5636    5691    5824    6505    6576
SKIP07          1847#   1986    2009    2031    6875    6941    7038
SLASH              1#   1250#
SROMCL          1835#   1913    1961    1972    3808    3820    4014    4017    4038    4041    4062    4065    4113    4116    4136
                4139    4159    4162    4211    4214    4236    4239    4261    4264    4313    4316    4337    4340    4361    4364
                4414    4417    4438    4441    4462    4465    4515    4518    4539    4542    4563    4566    4615    4618    4639
```

```
          4642    4663    4666    4715    4718    4739    4742    4763    4766    4815    4818    4841    4844    4865    4868
          4917    4920    4943    4946    4967    4970    5018    5021    5044    5047    5068    5071    5119    5122    5145
          5148    5169    5172    5221    5224    5247    5250    5271    5274    5322    5325    5348    5351    5372    5375
          5512    5618    5806    6890
STARS        1#   1250#
SVC          1#   1248#   1249
XFER         1#   1250#   3021#   3386#   3436#   3489#   3780#   3910#   4002#   4104#   4201#   4303#   4403#   4504#   4605#
          4705#   4805#   4907#   5009#   5110#   5212#   5313#   5412#   5493#   5588#   5776#   5893#   5967#   6173#   6423#
          6508#   6552#   6579#   6731#   6788#   6878#   6944#   7041#
XFERF        1#   1250#
XFERT        1#   1250#
$MD       2346#   2362    2391    2420    2449    2477    2505    2534    2563    2591    2619    2647    2675    2703    2732
          2762    2783    2811

. ABS.  030144      000

 ERRORS DETECTED:  0

 CZDMQE.BIN,CZDMQE.SEQ/SOL/CRF/NL:TOC=SVC34R.MLB,CZDMQE.P11
 RUN-TIME: 40 49 5 SECONDS
 RUN-TIME RATIO: 300/95=3.1
 CORE USED:  21K  (41 PAGES)
```