M8207,MCPU

M8207 STATIC DIAG. #1

CZDMPA0

AH-E226A-MC

FICHE 1 OF 2

SEP 1979

digital

MADE IN USA

M8207,MCPU

M8207 STATIC DIAG. #1

CZDMPA0

AH-E226A-MC

FICHE 2 OF 2

SEP 1979

digital

MADE IN USA

```
3298                    .REM @
3299
3300                                    IDENTIFICATION
3301                                    --------------
3302
3303                    PRODUCT CODE:   AC-E225A-MC
3304
3305                    PRODUCT NAME:   CZDMPA0 M8207 STATIC DIAG #1
3306
3307                    PRODUCT DATE:   MAY, 1979
3308
3309                    MAINTAINER:     DIAGNOSTICS MERRIMACK
3310
3311                    AUTHOR:         ED BADGER
3312
3313
3314
3315
3316
3317                            THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
3318                            NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
3319                            EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO
3320                            RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.
3321
3322                            NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF
3323                            SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS
3324                            AFFILIATED COMPANIES.
3325
3326                            COPYRIGHT (C) 1979 BY DIGITAL EQUIPMENT CORPORATION
3327
3328                    THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:
3329
3330                            DIGITAL         PDP             UNIBUS          MASSBUS
3331                            DEC             DECUS           DECTAPE
3332
```

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 FIRST OF TWO DIAGNOSTICS FOR THESE OPTIONS.

THE PROGRAM WAS IMPLEMENTED USING THE DIAGNOSTIC SUPERVISOR.

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

1.2 HARDWARE INTRODUCTION

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

2.0 HARDWARE REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE M8207 STATIC 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 PREVIOUS TO RUNNING THIS DIAGNOSTIC.

4.0 GENERAL PROGRAM CONSIDERATIONS

4.1 DIAGNOSTIC SUPERVISOR

THIS PROGRAM IS COMPATIBLE WITH THE STANDALONE DIAGNOSTIC

3442 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 30 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 ERROF 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 IDENTIFICATION AND 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
CZDMP-A-0
```

3556   M8207 DIAG.#1 OF 2

UNIT IS M8200,4,7
DR>

THE OPERATOR THEN PROCEEDES 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:

```
3615
3616
3617
3618
3619
3620                        HOE   HALT ON ERROR, CAUSING COMMAND MODE TO BE
3621                              ENTERED WHEN AN ERROR IS ENCOUNTERED
3622                        LOE   LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP
3623                              CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK
3624                              OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAIN-
3625                              ING THE ERROR
3626                        IER   INHIBIT ERROR REPORTING
3627                        IBE   INHIBIT BASIC ERROR REPORTS
3628                        IXE   INHIBIT EXTENDED ERROR REPORTS
3629                        PRI   DIRECT ALL MESSAGES TO A LINE PRINTER
3630                        PNT   PRINT NUMBER OF TEST BEING EXECUTED
3631                        BOE   BELL ON ERROR
3632                        UAM   RUN IN UNATTENDED MODE, BYPASSING MANUAL
3633                              INTERVENTION TESTS
3634                        ISR   INHIBIT STATISTICAL REPORTS
3635                        IDU   INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
3636                        LOT   LOOP ON TEST
3637
3638                  THE  FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0
3639                  ARE CLEARED.  A FLAG NOT SPECIFIED IS CLEARED.  IF THE FLAGS
3640                  SWITCH  IS  NOT GIVEN ALL FLAGS ARE CLEARED.  SEE EXAMPLE AT
3641                  END OF 6.3.1.5.
3642
3643
3644                  6.3.1.4 END OF PASS SWITCH (/EOP:<INCR>)
3645
3646                  <INCR> IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF
3647                  PASSES)  IT  IS  DESIRED  THAT  THE  END  OF PASS MESSAGE BE
3648                  PRINTED.  THE DEFAULT IS AT THE  END  OF  EVERY  PASS.   SEE
3649                  EXAMPLE AT END OF 6.3.1.5.
3650
3651
3652                  6.3.1.5 EFFECT OF START COMMAND
3653
3654                  THE  EFFECT OF THE START COMMAND IS TO INITIATE THE HARDWARE
3655                  PARAMETER DIALOGUE, THE  SOFTWARE  PARAMETER  DIALOGUE,  AND
3656                  THEN THE DIAGNOSTIC TESTS THEMSELVES.
3657
3658                  THE  HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION
3659                  "# UNITS?" TO WHICH THE  OPERATOR  REPLIES  WITH  A  DECIMAL
3660                  NUMBER  N FROM 1 TO 16. THE TERM 'UNIT" REFERS TO THE DEVICF
3661                  TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED.  FOLLOWING
3662                  THIS  ARE THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES WILL
3663                  BE BUILT.  EACH P-TABLE IS A CORE-RESIDENT TABLE  CONTAINING
3664                  ALL  THE  HARDWARE  INFORMATION  FOR ONE UNIT.  THE OPERATOR
3665                  MUST SUPPLY N (NUMBER OF UNITS) VALUES  FOR  EACH  QUESTION.
3666                  HE  MAY  DO  THIS  BY GIVING ONE ANSWER TO EACH QUESTION (IN
3667                  WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR
3668                  BY  GIVING  N  VALUES, SEPARATED BY COMMAS, TO EACH QUESTION
3669                  (SERIES WILL BE POSED ONCE).  EACH QUESTION IS  FOLLOWED  BY
3670                  THE  RESPONSE  RADIX  (D  FOR  DECIMAL,  B FOR BINARY, O FOR
```

```
3671                                    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

3729 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- INITAIL DIALOGUE (SEE 6.2),HARDWARE DIALOGUE (SEE 6.3.1.5), OR SOFTWARE DIALOGUE (SEE 6.3.1.5) CAUSES THE DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.

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

6.3.12 HARDWARE PARAMETERS

THE FOLLOWING 4 QUESTION 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 CARRIGE RETURN RESPONSE.

1. WHICH MICRO-PROCESSOR: (0) 7?

THE ALLOWABLE RESPONSES ARE 0 (M8200), 4 (M8204), AND THE DEFAULT 7 (M8207).

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

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

3. MICRO-PROCESSOR VECTOR ADDRESS: (O) 300?

THIS IS THE ADDRESS OF THE INPUT INTERRUPT VECTOR FOR THIS DEVICE. THE ALLOWABLE RANGE IS 000-674 (OCTAL), AND THE DEFAULT VALUE IS 300.

4. MICRO-PROCESSOR PRIORITY LEVEL: (O) 5?

THIS IS THE CPU PRIORITY AT WHICH THE INTERRUPT HANDLERS OF THE DEVICE WILL BE EXECUTED. THE ALLOWABLE RANGE IS 0-7, AND THE DEFAULT VALUE IS 5.

6.3.13 SOFTWARE PARAMETERS

NO SOFTWARE PARAMETER QUESTIONS ARE ASKED BY PART 1 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 7 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).

```
4066
4067
4068
4069
4070
4071                            7.0 TEST DESCRIPTION
4072
4073
4074
4075                            ******************** TEST 1 ********************
4076                            *VERIFY THAT REFERENCING UNIBUS DEVICE REGISTERS
4077                            *DOES NOT CAUSE A TIMEOUT TRAP
4078                            ************************************************
4079
4080
4081                            ******************** TEST 2 ********************
4082                            *VERIFY THAT RUN CAN BE CLEARED
4083                            ************************************************
4084
4085                            ******************** TEST 3 ********************
4086                            *UNIBUS REGISTER WORD DUAL ADDRESSING TEST
4087                            *LOAD ALL REGISTERS WITH INCREMENTING PATTERN
4088                            *READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING
4089                            *THE SEQUENCE:
4090                            *       1.  CLEAR REGISTER
4091                            *       2.  WRITE PATTERN
4092                            *       3.  VERIFY PATTERN
4093                            *       4.  DO ALL 4 REGISTERS
4094                            *       5.  READ ALL BACK IF ERRORS,
4095                            *           DUAL ADDRESS PROBLEM.
4096                            *
4097                            *           1 IN REG 0
4098                            *           2 IN REG 2
4099                            *           3 IN REG 4
4100                            *           4 IN REG 6
4101                            ************************************************
4102
4103
4104                            ******************** TEST 4 ********************
4105                            *CONTROL STATUS REGISTER WRITE/READ TEST
4106                            *FLOAT A ONE THROUGH BSEL 0
4107                            *CLEAR BIT0, VERIFY BIT0 WAS CLEARED
4108                            ************************************************
4109
4110
4111                            ******************** TEST 5 ********************
4112                            *CONTROL STATUS REGISTER WRITE/READ TEST
4113                            *SET BIT9, VERIFY BIT9 WAS SET
4114                            *CLEAR BIT9, VERIFY BIT9 WAS CLEARED
4115                            ************************************************
4116
4117
4118                            ******************** TEST 6 ********************
4119                            *CONTROL STATUS REGISTER WRITE/READ TEST
4120                            *SET BIT11, VERIFY BIT11 WAS SET
4121                            *CLEAR BIT11, VERIFY BIT11 WAS CLEARED
```

4122 **

```
4124
4125
4126
4127
4128
4129
4130
4131                    ********** TEST 7 **********
4132                    *CONTROL STATUS REGISTER WRITE/READ TEST
4133                    *SET BIT12, VERIFY BIT12 WAS SET
4134                    *CLEAR BIT 12, VERIFY BIT 12 WAS CLEARED
4135                    ************************************************
4136
4137
4138                    ********** TEST 8 **********
4139                    *CONTROL OUT REGISTER WRITE/READ TEST
4140                    *FLOAT A ONE THROUGH SEL2
4141                    ************************************************
4142
4143
4144                    ********** TEST 9 **********
4145                    *PORT4 REGISTER WRITE/READ TEST
4146                    *FLOAT A ONE THROUGH PORT4 REGISTER
4147                    *FLOAT A ZERO THROUGH PORT4 REGISTER
4148                    ************************************************
4149
4150
4151                    ********** TEST 10 **********
4152                    *PORT6 REGISTER WRITE/READ TEST
4153                    *FLOAT A ONE THROUGH PORT6 REGISTER
4154                    *FLOAT A ZERO THROUGH PORT6 REGISTER
4155                    ************************************************
4156
4157
4158                    ********** TEST 11 **********
4159                    *UNIBUS REGISTER BYTE DUAL ADDRESSING TEST
4160                    *LOAD ALL REGISTERS WITH INCREMENTING PATTERN
4161                    *READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING
4162                    ************************************************
4163
4164
4165                    ********** TEST 12 **********
4166                    *MAINTENANCE INSTRUCTION REGISTER TEST
4167                    *VERIFY THAT THE MAINT IR CAN BE WRITTEN TO ALL ZEROS'
4168                    *AND ALL ONES'.  VERIFY THAT IS IS CLEARED ON A BUS RESET.
4169                    ************************************************
4170
4171
4172                    ********** TEST 13 **********
4173                    *MICRO PROCESSOR TEST
4174                    *LOAD KMP06 WITH A MICRO-PROCESSOR INSTRUCTION, CLOCK IT
4175                    *VERIFY INSTRUCTION EXECUTED PROPERLY
4176                    *INSTRUCTION SHOULD MOVE IBUS*4 TO IBUS*5, IBUS*4 IS ALL 1'S
4177                    *AND IBUS*5 IS ALL 0'S. RESULT SHOULD BE ALL 1'S IN SEL4
4178                    ************************************************
4179
```

```
4180
```

```
4182
4183
4184
4185
4186
4187                    ********** TEST 14 **********
4188                    *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
4189                    *FLOAT A 1 THROUGH IBUS* REGISTER 0
4190                    *FLOAT A 0 THROUGH IBUS* REGISTER 0
4191                    **************************************************
4192
4193
4194                    ********** TEST 15 **********
4195                    *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
4196                    *FLOAT A 1 THROUGH IBUS* REGISTER 2
4197                    *FLOAT A 0 THROUGH IBUS* REGISTER 2
4198                    **************************************************
4199
4200
4201                    ********** TEST 16 **********
4202                    *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
4203                    *FLOAT A 1 THROUGH IBUS* REGISTER 4
4204                    *FLOAT A 0 THROUGH IBUS* REGISTER 4
4205                    **************************************************
4206
4207
4208                    ********** TEST 17 **********
4209                    *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
4210                    *FLOAT A 1 THROUGH IBUS* REGISTER 5
4211                    *FLOAT A 0 THROUGH IBUS* REGISTER 5
4212                    **************************************************
4213
4214
4215                    ********** TEST 18 **********
4216                    *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
4217                    *FLOAT A 1 THROUGH IBUS* REGISTER 10
4218                    *FLOAT A 0 THROUGH IBUS* REGISTER 10
4219                    **************************************************
4220
4221
4222                    ********** TEST 19 **********
4223                    *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
4224                    *FLOAT A 1 THROUGH IBUS* REGISTER 11
4225                    *FLOAT A 0 THROUGH IBUS* REGISTER 11
4226                    **************************************************
4227
4228
4229                    ********** TEST 20 **********
4230                    *MICRO PROCESSOR IBUS REBISTER WRITE/READ TEST
4231                    *FLOAT A 1 THROUGH IBUS REGISTER 0
4232                    *FLOAT A 0 THROUGH IBUS REGISTER 0
4233                    **************************************************
4234
4235
4236                    ********** TEST 21 **********
4237                    *MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
```

```
4238                                    *FLOAT A 1 THROUGH IBUS REGISTER 1
```

```
4240
4241
4242
4243
4244
4245                    *FLOAT A 0 THROUGH IBUS REGISTER 1
4246                    ************************************************
4247
4248
4249                    ********************* TEST 22 *****************
4250                    *MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
4251                    *FLOAT A 1 THROUGH IBUS REGISTER 2
4252                    *FLOAT A 0 THROUGH IBUS REGISTER 2
4253                    ************************************************
4254
4255
4256                    ********************* TEST 23 *****************
4257                    *MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
4258                    *FLOAT A 1 THROUGH IBUS REGISTER 3
4259                    *FLOAT A 0 THROUGH IBUS REGISTER 3
4260                    ************************************************
4261
4262
4263                    ********************* TEST 24 *****************
4264                    *MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
4265                    *FLOAT A 1 THROUGH IBUS REGISTER 4
4266                    *FLOAT A 0 THROUGH IBUS REGISTER 4
4267                    ************************************************
4268
4269
4270                    ********************* TEST 25 *****************
4271                    *MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
4272                    *FLOAT A 1 THROUGH IBUS REGISTER 5
4273                    *FLOAT A 0 THROUGH IBUS REGISTER 5
4274                    ************************************************
4275
4276
4277                    ********************* TEST 26 *****************
4278                    *MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
4279                    *FLOAT A 1 THROUGH IBUS REGISTER 6
4280                    *FLOAT A 0 THROUGH IBUS REGISTER 6
4281                    ************************************************
4282
4283
4284                    ********************* TEST 27 *****************
4285                    *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
4286                    *FLOAT A 1 THROUGH IBUS* REGISTER 7
4287                    *FLOAT A 0 THROUGH IBUS* REGISTER 7
4288                    ************************************************
4289
4290
4291                    ********************* TEST 28 *****************
4292                    *MICRO PROCESSOR IBUS DUAL ADDRESS TEST
4293                    *WRITE ALL IBUS REGISTERS WITH INCREMENTING PATTERN
4294                    *READ ALL IBUS REGISTERS TO VERIFY CORRECT ADDRESSING
4295                    ************************************************
```

4296

```
4298
4299
4300
4301
4302
4303
4304                    ********** TEST 29 **********
4305                    *MICRO PROCESSOR BR REGISTER TEST
4306                    *FLOAT A 1 THROUGH THE BR
4307                    *FLOAT A 0 THROUGH THE BR
4308                    ************************************************
4309
4310
4311                    ********** TEST 30 **********
4312                    *SCRATCH PAD TEST
4313                    *FLOAT A 1 THROUGH EACH SCRATCH PAD LOCATION
4314                    *FLOAT A 0 THROUGH EACH SCRATCH PAD LOCATION
4315                    ************************************************
4316
4317
4318                    ********** TEST 31 **********
4319                    *SCRATCH PAD DUAL ADDRESSING TEST
4320                    *WRITE AN INCREMENTING PATTERN IN ALL SP LOCATIONS
4321                    *READ ALL SP LOCATIONS TO VERIFY CORRECT ADDRESSING
4322                    ************************************************
4323
4324
4325                    ********** TEST 32 **********
4326                    *INTERRUPT TEST
4327                    *TEST THAT DEVICE CAN INTERRUPT TO VECTOR A
4328                    ************************************************
4329
4330
4331                    ********** TEST 33 **********
4332                    *INTERRUPT TEST
4333                    *TEST THAT DEVICE CAN INTERRUPT TO VECTOR B
4334                    ************************************************
4335
4336
4337                    ********** TEST 34 **********
4338                    *PRIORITY INTERRUPT TEST
4339                    *SET PS TO ALL BR LEVELS EQUAL OR GREATER THAN
4340                    *THE M8200,4,7 LEVEL, VERIFY THAT M8200,4,7 DOES NOT INTERRUPT
4341                    ********** **********
4342
4343
4344                    ********** TEST 35 **********
4345                    *PRIORITY INTERRUPT TESTS
4346                    *SET PS TO ALL BR LEVELS LESS THAN THE M8200,4,7 LEVEL
4347                    *VERIFY THAT ALL M8200,4,7 WILL INTERRUPT
4348                    ************************************************
4349
4350
4351                    ********** TEST 36 **********
4352                    *NPR TEST
4353                    *TEST OF DATO, 1 WORD FROM UPROC TO 11 MEMORY
```

```
 4354                                  **************************************************
```

```
4356
4357
4358
4359
4360
4361
4362
4363                    ********** TEST 37 **********
4364                    *NPR TEST
4365                    *TEST OF DATI, 1 WORD FROM 11 MEMORY TO UPROC
4366                    ************************************************
4367
4368
4369                    ********** TEST 38 **********
4370                    *NPR TEST
4371                    *TEST OF DATOB, 1 BYTE FROM UPROC TO 11 MEMORY
4372                    ************************************************
4373
4374
4375                    ********** TEST 39 **********
4376                    *TEST OF EA BITS 16 AND 17
4377                    *DO A DATO TO AN ADDRESS USING OUT BA BITS 16 AND 17
4378                    *VERIFY CORRECT RESULTS
4379                    ************************************************
4380
4381
4382                    ********** TEST 40 **********
4383                    *TEST OF EA BITS 16 AND 17
4384                    *DO A DATI USING IN BA BITS 16 AND 17
4385                    *VERIFY CORRECT RESULTS
4386                    *IN ORDER TO DO THIS TEST, WE WILL READ THE DATA FROM THE
4387                    *CONSOL TTY CSR IF ONE EXISTS
4388                    *IF NO COSOL TTY CSR AT ADDRESS 177560, THIS TEST
4389                    *WILL BE SKIPPED
4390                    ************************************************
4391
4392
4393                    ********** TEST 41 **********
4394                    *NPR NON-EXISTENT MEMORY TEST
4395                    *DO A DATO TO A NON-EXISTENT ADDRESS
4396                    *VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
4397                    ************************************************
4398
4399
4400                    ********** TEST 42 **********
4401                    *NPR NON-EXISTENT MEMORY TEST
4402                    *DO A DATI FROM A NON-EXISTENT ADDRESS
4403                    *VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
4404                    ************************************************
4405
4406
4407                    ********** TEST 43 **********
4408                    *NPR TEST
4409                    *USING DATO, NPR A BINARY COUNT (0-377)
4410                    *FROM MICRO-PROCESSOR TO ALL AVAILABLE MEMORY
4411                    ************************************************
```

E 3

4412

```
4414
4415
4416
4417
4418
4419
4420                    ********** TEST 44 **********
4421                    *ALU C BIT TEST
4422                    *TEST THAT AN ADD OF 377 AND 377 WILL SET THE C BIT
4423                    ****************************************************
4424
4425
4426                    ********** TEST 45 **********
4427                    *ALU TEST
4428                    *TEST OF ALU FUNCTION SEL B WITH C BIT CLEARED
4429                    *ALU FUNCTION (B)    CODE=11
4430                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4431                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4432                    ****************************************************
4433
4434
4435                    ********** TEST 46 **********
4436                    *ALU TEST
4437                    *TEST OF ALU FUNCTION SEL A WITH C BIT CLEARED
4438                    *ALU FUNCTION (A)    CODE=10
4439                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4440                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4441                    ****************************************************
4442
4443
4444                    ********** TEST 47 **********
4445                    *ALU TEST
4446                    *TEST OF ALU FUNCTION A OR NOTB WITH C BIT CLEARED
4447                    *ALU FUNCTION (A OR NOTB)    CODE=12
4448                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4449                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4450                    ****************************************************
4451
4452
4453                    ********** TEST 48 **********
4454                    *ALU TEST
4455                    *TEST OF ALU FUNCTION A AND B  WITH C BIT CLEARED
4456                    *ALU FUNCTION (A AND B)    CODE=13
4457                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4458                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4459                    ****************************************************
4460
4461
4462                    ********** TEST 49 **********
4463                    *ALU TEST
4464                    *TEST OF ALU FUNCTION A OR B WITH C BIT CLEARED
4465                    *ALU FUNCTION  (A OR B)    CODE=14
4466                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4467                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4468                    ****************************************************
4469
```

4470

```
4472
4473
4474
4475
4476
4477                    ********** TEST 50 **********
4478                    *ALU TEST
4479                    *TEST OF ALU FUNCTION A XOR B WITH C BIT
4480                    *ALU FUNCTION (A XOR B)    CODE=15
4481                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4482                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4483                    ************************************************
4484
4485
4486                    ********** TEST 51 **********
4487                    *ALU TEST
4488                    *TEST OF ALU FUNCTION ADD WITH C BIT CLEARED
4489                    *ALU FUNCTION (A PLUS B)    CODE=00
4490                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4491                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4492                    ************************************************
4493
4494
4495                    ********** TEST 52 **********
4496                    *ALU TEST
4497                    *TEST OF ALU FUNCTION 2A W.C WITH C BIT CLEARED
4498                    *ALU FUNCTION (A PLUS A PLUS C)    CODE=6
4499                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4500                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4501                    ************************************************
4502
4503
4504                    ********** TEST 53 **********
4505                    *ALU TEST
4506                    *TEST OF ALU FUNCTION SUB WITH C BIT CLEARED
4507                    *ALU FUNCTION (A-B)    CODE=16
4508                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4509                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4510                    ************************************************
4511
4512
4513                    ********** TEST 54 **********
4514                    *ALU TEST
4515                    *TEST OF ALU FUNCTION ADD W/C WITH C BIT CLEARED
4516                    *ALU FUNCTION (A PLUS B PLUS C)    CODE=01
4517                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4518                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4519                    ************************************************
4520
4521
4522                    ********** TEST 55 **********
4523                    *ALU TEST
4524                    *TEST OF ALU FUNCTION SUB W/C WITH C BIT CLEARED
4525                    *ALU FUNCTION (A-B-C)    CODE=2
4526                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4527                    *PERFORM THE FUNCTION. VERIFY THE RESULTS
```

```
4528                    ****************************************************
```

```
4530
4531
4532
4533
4534
4535
4536
4537                    ********** TEST 56 **********
4538                    *ALU TEST
4539                    *TEST OF ALU FUNCTION INC A WITH C BIT CLEARED
4540                    *ALU FUNCTION (A PLUS 1)    CODE =3
4541                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4542                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4543                    ************************************************
4544
4545
4546                    ********** TEST 57 **********
4547                    *ALU TEST
4548                    *TEST OF ALU FUNCTION 2A WITH C BIT CLEARED
4549                    *ALU FUNCTION (A PLUS A)    CODE=5
4550                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4551                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4552                    ************************************************
4553
4554
4555                    ********** TEST 58 **********
4556                    *ALU TEST
4557                    *TEST OF ALU FUNCTION A PLUS C WITH C BIT CLEARED
4558                    *ALU FUNCTION (A PLUS C)    CODE=4
4559                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4560                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4561                    ************************************************
4562
4563
4564                    ********** TEST 59 **********
4565                    *ALU TEST
4566                    *TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT CLEARED
4567                    *ALU FUNCTION (A-B-1)    CODE=17
4568                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4569                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4570                    ************************************************
4571
4572
4573                    ********** TEST 60 **********
4574                    *ALU TEST
4575                    *TEST OF ALU FUNCTION DEC A WITH C BIT CLEARED
4576                    *ALU FUNCTION (A-1)    CODE=7
4577                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4578                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4579                    ************************************************
4580
4581
4582                    ********** TEST 61 **********
4583                    *ALU TEST
4584                    *TEST OF ALU FUNCTION SEL B WITH C BIT SET
4585                    *ALU FUNCTION (B)    CODE=11
```

```
4586                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
```

```
4588
4589
4590
4591
4592
4593                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4594                    ****************************************************
4595
4596
4597                    ******************** TEST 62 **********************
4598                    *ALU TEST
4599                    *TEST OF ALU FUNCTION SEL A WITH C BIT SET
4600                    *ALU FUNCTION (A)    CODE=10
4601                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4602                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4603                    ****************************************************
4604
4605
4606                    ******************** TEST 63 **********************
4607                    *ALU TEST
4608                    *TEST OF ALU FUNCTION A OR NOTB WITH C BIT SET
4609                    *ALU FUNCTION (A OR NOTB)    CODE=12
4610                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4611                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4612                    ****************************************************
4613
4614
4615                    ******************** TEST 64 **********************
4616                    *ALU TEST
4617                    *TEST OF ALU FUNCTION A AND B WITH C BIT SET
4618                    *ALU FUNCTION (A AND B)    CODE=13
4619                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4620                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4621                    ****************************************************
4622
4623
4624                    ******************** TEST 65 **********************
4625                    *ALU TEST
4626                    *TEST OF ALU FUNCTION A OR B WITH C BIT SET
4627                    *ALU FUNCTION (A OR B)    CODE=14
4628                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4629                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4630                    ****************************************************
4631
4632
4633                    ******************** TEST 66 **********************
4634                    *ALU TEST
4635                    *TEST OF ALU FUNCTION A XOR B WITH C BIT SET
4636                    *ALU FUNCTION (A XOR B)    CODE=15
4637                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4638                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4639                    ****************************************************
4640
4641
4642                    ******************** TEST 67 **********************
4643                    *ALU TEST
```

```
4644                                    *TEST OF ALU FUNCTION ADD WITH C BIT SET
```

```
4646
4647
4648
4649
4650
4651                    *ALU FUNCTION (A PLUS B)      CODE=00
4652                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4653                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4654                    ****************************************************
4655
4656
4657                    ******************** TEST 68 ***********************
4658                    *ALU TEST
4659                    *TEST OF ALU FUNCTION 2A W/C WITH C BIT SET
4660                    *ALU FUNCTION (A PLUS A PLUS C)     CODE=6
4661                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4662                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4663                    ****************************************************
4664
4665
4666                    ******************** TEST 69 ***********************
4667                    *ALU TEST
4668                    *TEST OF ALU FUNCTION SUB WITH C BIT SET
4669                    *ALU FUNCTION (A-B)     CODE=16
4670                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4671                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4672                    ****************************************************
4673
4674
4675                    ******************** TEST 70 ***********************
4676                    *ALU TEST
4677                    *TEST OF ALU FUNCTION ADD W/C WITH C BIT SET
4678                    *ALU FUNCTION (A PLUS B PLUS C)     CODE=01
4679                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4680                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4681                    ****************************************************
4682
4683
4684                    ******************** TEST 71 ***********************
4685                    *ALU TEST
4686                    *TEST OF ALU FUNCTION SUB W/C WITH C BIT SET
4687                    *ALU FUNCTION (A-B-C)     CODE=2
4688                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4689                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4690                    ****************************************************
4691
4692
4693                    ******************** TEST 72 ***********************
4694                    *ALU TEST
4695                    *TEST OF ALU FUNCTION INC A WITH C BIT SET
4696                    *ALU FUNCTION (A PLUS 1)     CODE=3
4697                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4698                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4699                    ****************************************************
4700
4701
```

4702 ********** TEST 73 **********

```
4704
4705
4706
4707
4708
4709                    *ALU TEST
4710                    *TEST OF ALU FUNCTION 2A WITH C BIT SET
4711                    *ALU FUNCTION (A PLUS A)    CODE=5
4712                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4713                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4714                    ********************************************************
4715
4716
4717                    ******************** TEST 74 **********************
4718                    *ALU TEST
4719                    *TEST OF ALU FUNCTION A PLUS C WITH C BIT SET
4720                    *ALU FUNCTION (A PLUS C)    CODE=4
4721                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4722                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4723                    ********************************************************
4724
4725
4726                    ******************** TEST 75 **********************
4727                    *ALU TEST
4728                    *TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT SET
4729                    *ALU FUNCTION (A-B-1)    CODE=17
4730                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4731                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
4732                    ********************************************************
4733
4734
4735                    ******************** TEST 76 **********************
4736                    *ALU TEST
4737                    *TEST OF ALU FUNCTION DEC A WITH C BIT SET
4738                    *ALU FUNCTION (A-1)    CODE=7
4739                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4740                    *PERFORM THE FUNCTION, VERIFY THE RESULT
4741                    ********************************************************
4742
4743
4744
```

```
4746
4747
4748
4749
4750
4751                    8.0 ERROR INFORMATION
4752
4753
4754                    8.1 ERROR REPORTING
4755
4756                    ERRORS ARE REPORTED BY THE PROGRAM AS THEY OCCUR (IF NOT
4757                    INHIBITED). THE REPORT CONFORMS TO THE DIAGNOSTIC SUPERVISOR
4758                    ERROR REPORT FORMAT, AND CONSISTS OF A DESCRIPTION OF THE
4759                    ERROR, THE TEST NUMBER, SUBTEST NUMBER, PC OF THE ERROR
4760                    CALL, DEVICE ADDRESS, AND BASIC AND EXTENDED ERROR
4761                    INFORMATION.
4762
4763
4764
4765                    THE FOLLOWING EXAMPLE PROVIDES A TYPICAL ERROR REPORT.
4766
4767
4768                    CZDMP DVC FTL ERR 00003 TST 029 SUB 000 PC:02<626
4769
4770                    BR REGISTER DATA TEST
4771                    UNIT=00; FAILING UNIT ADDRESS=160170
4772
4773                    GOOD            BAD
4774                    177776          000011
4775
4776
4777
4778                    FOR ALL OTHER ERRORS, THE REPORT MAY BE MORE EXTENSIVE AND
4779                    REQUIRE ADDITIONAL DATA TO BE REPORTED.
4780
4781
4782
4783                    @
```

```
4785
4786
4787
4788
4789
```

```
                                .TITLE CZDMPA0 M8207 STATIC DIAG #1
                002000                  .=2000





                                        .MCALL  SVC
        002000                          SVC                                     ; INITIALIZE SUPERVISOR MACROS





        002000                          BGNMOD  CZDMP


                000000          $LSTIN= 0
                000000          $LSTTAG  0
                000000          SVCINS= 0       ; LIST INSTRUCTIONS, SHIFTED RIGHT
                000000          SVCTST= 0       ; LIST TEST TAGS, SHIFTED RIGHT
                000000          SVCSUB= 0       ; LIST SUBTEST TAGS, SHIFTED RIGHT
                000000          SVCGBL= 0       ; LIST GLOBAL TAGS, SHIFTED RIGHT
                000000          SVCTAG= 0       ; LIST OTHER TAGS, SHIFTED RIGHT

                                ;       CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
                                ;       TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS.  CHANGE THE
                                ;       SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS.  YOU MAY
                                ;       CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.


                                .ENABL  AMA
```

```
 4832                                   .SBTTL  PROGRAM HEADER
 4833                                   ;++
 4834                                   ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
 4835                                   ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
 4836                                   ;--
 4837
 4838  002000                                   POINTER BGNAU,BGNDU
 4839
 4847
 4848  002000                                   HEADER  CZDMP,A,0,120.,0
  (4)  002000                           L$NAME::                ;DIAGNOSTIC NAME
  (4)  002000      103                          .ASCII /C/
  (4)  002001      132                          .ASCII /Z/
  (4)  002002      104                          .ASCII /D/
  (4)  002003      115                          .ASCII /M/
  (4)  002004      120                          .ASCII /P/
  (6)  002005      000                          .BYTE   0
  (6)  002006      000                          .BYTE   0
  (5)  002007      000                          .BYTE   0
  (5)  002010                           L$REV::                 ;REVISION LEVEL
  (4)  002010      101                          .ASCII  /A/
  (5)  002011                           L$DEPO::                ;0
  (4)  002011      060                          .ASCII  /0/
  (5)  002012                           L$UNIT::                ;NUMBER OF UNITS
  (4)  002012  000000                           .WORD   0
  (5)  002014                           L$TIML::                ;LONGEST TEST TIME
  (4)  002014  000170                           .WORD   120.
  (5)  002016                           L$HPCP::                ;POINTER TO H.W. QUES.
  (4)  002016  034704                           .WORD   L$HARD
  (5)  002020                           L$SPCP::                ;POINTER TO S.W. QUES.
  (4)  002020  000000                           .WORD   0
  (5)  002022                           L$HPTP::                ;PTR. TO DEF. H.W. PTABLE
  (4)  002022  002364                           .WORD   L$HW
  (5)  002024                           L$SPTP::                ;PTR. TO S.W. PTABLE
  (4)  002024  000000                           .WORD   0
  (5)  002026                           L$LADP::                ;DIAG. END ADDRESS
  (4)  002026  040004                           .WORD   L$LAST
  (5)  002030                           L$STA::                 ;RESERVED FOR APT STATS
  (4)  002030  000000                           .WORD   0
  (5)  002032                           L$CO::
  (4)  002032  000000                           .WORD   0
  (5)  002034                           L$DTYP::                ;DIAGNOSTIC TYPE
  (4)  002034  000000                           .WORD   0
  (5)  002036                           L$APT::                 ;APT EXPANSION
  (4)  002036  000000                           .WORD   0
  (5)  002040                           L$DTP::                 ;PTR. TO DISPATCH TABLE
  (4)  002040  002132                           .WORD   L$DISPATCH
  (5)  002042                           L$EXP1::                ;EXPANSION WORDS
  (4)  002042  000000                           .WORD   0
  (5)  002044                           L$EXP2::
  (4)  002044  000000                           .WORD   0
  (5)  002046                           L$EXP3::
  (4)  002046  000000                           .WORD   0
  (5)  002050                           L$MREV::                ;SVC REV AND EDIT #
  (4)  002050      003                          .BYTE   C$REVISION
  (3)  002051      000                          .BYTE   C$EDIT
```

```
(5)  002052
(4)  002052  000000
(5)  002054  000000
(5)  002056
(4)  002056  000000
(5)  002060
(4)  002060  003130
(5)  002062
(4)  002062  000000
(5)  002064
(4)  002064  000000
(5)  002066
(4)  002066  000000
(5)  002070
(4)  002070  011350
(5)  002072
(4)  002072  011344
(5)  002074
(4)  002074  000000
(5)  002076
(4)  002076  002414
(5)  002100
(4)  002100  104035
(5)  002102
(4)  002102  000000
(5)  002104
(4)  002104  010554
(5)  002106
(4)  002106  011340
(5)  002110
(4)  002110  011242
(5)  002112
(4)  002112  002122
(5)  002114
(4)  002114  000000
(5)  002116
(4)  002116  000000
(5)  002120
(4)  002120  000000
4849
4850
4856
4857  002122
 (3)  002122
4858  002122  177777
4859  002124  177777
4860  002126  177777
4861  002130
4862

L$EF::                          ;DIAG. EVENT FLAGS
        .WORD   0
        .WORD   0
L$SPC::
        .WORD   0
L$DEVP::                        ; POINTER TO DEVICE TYPE LIST
        .WORD   L$DVTYP
L$REPP::                        ;PTR. TO REPORT CODE
        .WORD   0
L$EXP4::
        .WORD   0
L$EXP5::
        .WORD   0
L$AUT::                         ;PTR. TO ADD UNIT CODE
        .WORD   L$AU
L$DUT::                         ;PTR. TO DROP UNIT CODE
        .WORD   L$DU
L$LUN::                         ; LUN FOR EXERCISERS TO FILL
        .WORD   0
L$DESP::                        ; POINTER TO DIAG. DESCRIPTION
        .WORD   L$DESC
L$LOAD::                        ;GENERATE SPECIAL AUTOLOAD EMT
        EMT     E$LOAD
L$ETP::                         ;POINTER TO ERRTBL
        .WORD   0
L$ICP::                         ;PTR. TO INIT CODE
        .WORD   L$INIT
L$CCP::                         ;PTR. TO CLEAN-UP CODE
        .WORD   L$CLEAN
L$ACP::                         ;PTR. TO AUTO CODE
        .WORD   L$AUTO
L$PRT::                         ;PTR. TO PROTECT TABLE
        .WORD   L$PROT
L$TEST::                        ;TEST NUMBER
        .WORD   0
L$DLY::                         ;DELAY COUNT
        .WORD   0
L$HIME::                        ;PTR. TO HIGH MEM
        .WORD   0


        BGNPROT
L$PROT::
        .WORD   -1
        .WORD   -1
        .WORD   -1
        ENDPROT
```

```
 4864
 4865                                   .SBTTL  DISPATCH TABLE
 4866
 4867                                   ;////////////////////////////////////////////////////////////////////////////
 4868                                   ;/ THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
 4869                                   ;/ IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
 4870                                   ;////////////////////////////////////////////////////////////////////////////
 4871  002130                                   DISPATCH 76.
  (4)  002130  000114                           .WORD   76
  (3)  002132                           L$DISPATCH::
  (6)  002132  011352                           .WORD   T1
  (6)  002134  011500                           .WORD   T2
  (6)  002136  011544                           .WORD   T3
  (6)  002140  011732                           .WORD   T4
  (6)  002142  012076                           .WORD   T5
  (6)  002144  012232                           .WORD   T6
  (6)  002146  012362                           .WORD   T7
  (6)  002150  012512                           .WORD   T8
  (6)  002152  012654                           .WORD   T9
  (6)  002154  013040                           .WORD   T10
  (6)  002156  013224                           .WORD   T11
  (6)  002160  013412                           .WORD   T12
  (6)  002162  013562                           .WORD   T13
  (6)  002164  013670                           .WORD   T14
  (6)  002166  014120                           .WORD   T15
  (6)  002170  014350                           .WORD   T16
  (6)  002172  014600                           .WORD   T17
  (6)  002174  015030                           .WORD   T18
  (6)  002176  015314                           .WORD   T19
  (6)  002200  015610                           .WORD   T20
  (6)  002202  016040                           .WORD   T21
  (6)  002204  016270                           .WORD   T22
  (6)  002206  016520                           .WORD   T23
  (6)  002210  016750                           .WORD   T24
  (6)  002212  017200                           .WORD   T25
  (6)  002214  017430                           .WORD   T26
  (6)  002216  017660                           .WORD   T27
  (6)  002220  020110                           .WORD   T28
  (6)  002222  020406                           .WORD   T29
  (6)  002224  020726                           .WORD   T30
  (6)  002226  021272                           .WORD   T31
  (6)  002230  021604                           .WORD   T32
  (6)  002232  021746                           .WORD   T33
  (6)  002234  022110                           .WORD   T34
  (6)  002236  022264                           .WORD   T35
  (6)  002240  022470                           .WORD   T36
  (6)  002242  022634                           .WORD   T37
  (6)  002244  023004                           .WORD   T38
  (6)  002246  023150                           .WORD   T39
  (6)  002250  023336                           .WORD   T40
  (6)  002252  023552                           .WORD   T41
  (6)  002254  023770                           .WORD   T42
  (6)  002256  024122                           .WORD   T43
  (6)  002260  024340                           .WORD   T44
  (6)  002262  024500                           .WORD   T45
  (6)  002264  024704                           .WORD   T46
```

```
   (6)  002266  025110                    .WORD   T47
   (6)  002270  025314                    .WORD   T48
   (6)  002272  025520                    .WORD   T49
   (6)  002274  025724                    .WORD   T50
   (6)  002276  026130                    .WORD   T51
   (6)  002300  026334                    .WORD   T52
   (6)  002302  026540                    .WORD   T53
   (6)  002304  026746                    .WORD   T54
   (6)  002306  027152                    .WORD   T55
   (6)  002310  027356                    .WORD   T56
   (6)  002312  027562                    .WORD   T57
   (6)  002314  027766                    .WORD   T58
   (6)  002316  030172                    .WORD   T59
   (6)  002320  030376                    .WORD   T60
   (6)  002322  030602                    .WORD   T61
   (6)  002324  031006                    .WORD   T62
   (6)  002326  031212                    .WORD   T63
   (6)  002330  031416                    .WORD   T64
   (6)  002332  031622                    .WORD   T65
   (6)  002334  032026                    .WORD   T66
   (6)  002336  032232                    .WORD   T67
   (6)  002340  032436                    .WORD   T68
   (6)  002342  032642                    .WORD   T69
   (6)  002344  033046                    .WORD   T70
   (6)  002346  033252                    .WORD   T71
   (6)  002350  033456                    .WORD   T72
   (6)  002352  033662                    .WORD   T73
   (6)  002354  034066                    .WORD   T74
   (6)  002356  034272                    .WORD   T75
   (6)  002360  034476                    .WORD   T76
 4872                                     ;LNT.ED DIFINED AT END OF PROGRAM TO BE LAST TEST NUMBER.
 4873
 4880
 4881
 4882
 4883
 4884
```

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

002362                                  BGNHW   DFPTBL
(3)  002362  000013                     .WORD   L10001-L$HW/2
(3)  002364                     L$HW::
(3)  002364                     DFPTBL::

002364  000007                          .WORD   7               ;MICRO CPU TYPE
002366  160170                          .WORD   160170          ;M8200,4,7 CSR ADDRESS
002370  000300                          .WORD   300             ;M8200,4,7 VECTOR ADDRESS
002372  005000                          .WORD   5000            ;INTERRUPT PRIORITY LEVEL
002374  000003                          .WORD   3               ;LINE UNIT TYPE
002376  000056                          .WORD   56              ;SWITCH PACK #1 (DDCMP LINE #)
002400  000000                          .WORD   0               ;SWITCH PACK #2 (BM873 BOOT ADDRESS)
002402  000000                          .WORD   0               ;SWITCH PACK #3
002404  000000                          .WORD   0               ;TEST CONNECTOR INSTALLED FLAG
002406  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
002410  000000                          .WORD   0               ;0=RUN SW OFF, 1=SW ON
002412                                  ENDHW
(3)  002412                     L10001:
```

```
                                .SBTTL  SOFTWARE P-TABLE

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

002412                                  BGNSW   SFPTBL
(3) 002412  000000              .       .WORD   L10002-L$SW/2
(3) 002414                      L$SW::
(3) 002414                      SFPTBL::

002414                                  ENDSW
(3) 002414                      L10002:
```

```
 4934
 4935
 4936
 4937
 4938
 4939
 4940
 4941
 4942
 4943
 4944
 4945                                   .SBTTL  GLOBAL EQUATES SECTION
 4946
 4947
 4948                                   ;//////////////////////////////////////////////////////////////////////////////
 4949                                   ;/      THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
 4950                                   ;/      ARE USED IN MORE THAN ONE TEST.
 4951                                   ;//////////////////////////////////////////////////////////////////////////////
 4952
 4953  002414                                   EQUALS
  (1)                                   ;
  (1)                                   ; BIT DIFINITIONS
  (1)                                   ;
  (1)           100000                  BIT15== 100000
  (1)           040000                  BIT14== 40000
  (1)           020000                  BIT13== 20000
  (1)           010000                  BIT12== 10000
  (1)           004000                  BIT11== 4000
  (1)           002000                  BIT10== 2000
  (1)           001000                  BIT09== 1000
  (1)           000400                  BIT08== 400
  (1)           000200                  BIT07== 200
  (1)           000100                  BIT06== 100
  (1)           000040                  BIT05== 40
  (1)           000020                  BIT04== 20
  (1)           000010                  BIT03== 10
  (1)           000004                  BIT02== 4
  (1)           000002                  BIT01== 2
  (1)           000001                  BIT00== 1
  (1)                                   ;
  (1)           001000                  BIT9==  BIT09
  (1)           000400                  BIT8==  BIT08
  (1)           000200                  BIT7==  BIT07
  (1)           000100                  BIT6==  BIT06
  (1)           000040                  BIT5==  BIT05
  (1)           000020                  BIT4==  BIT04
  (1)           000010                  BIT3==  BIT03
  (1)           000004                  BIT2==  BIT02
  (1)           000002                  BIT1==  BIT01
  (1)           000001                  BIT0==  BIT00
  (1)                                   ;
  (1)                                   ; EVENT FLAG DEFINITIONS
  (1)                                   ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
  (1)                                   ;
  (1)           000040                  EF.START==      32.                     ; START COMMAND WAS ISSUED
  (1)           000037                  EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
```

```
   (1)          000036                  EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
   (1)          000035                  EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
   (1)          000034                  EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
   (1)                                  ;
   (1)                                  ;
   (1)                                  ; PRIORITY LEVEL DEFINITIONS
   (1)                                  ;
   (1)          000340                  PRI07== 340
   (1)          000300                  PRI06== 300
   (1)          000240                  PRI05== 240
   (1)          000200                  PRI04== 200
   (1)          000140                  PRI03== 140
   (1)          000100                  PRI02== 100
   (1)          000040                  PRI01== 40
   (1)          000000                  PRI00== 0
   (1)                                  ;
   (1)                                  ;OPERATOR FLAG BITS
   (1)                                  ;
   (1)          000004                  EVL==       4
   (1)          000010                  LOT==      10
   (1)          000020                  ADR==      20
   (1)          000040                  IDU==      40
   (1)          000100                  ISR==     100
   (1)          000200                  UAM==     200
   (1)          000400                  BOE==     400
   (1)          001000                  PNT==    1000
   (1)          002000                  PRI==    2000
   (1)          004000                  IXE==    4000
   (1)          010000                  IBE==   10000
   (1)          020000                  IER==   20000
   (1)          040000                  LOE==   40000
   (1)          100000                  HCE==  100000
4954
4955
4956
4957
4958
4959                                     ;*****************************************************************************
4960                                     ;* INSTRUCTION DEFINITIONS
4961                                     ;*****************************************************************************
4962          022626                     POP2SP=22626            ;INCREMENT STACK TWICE
4963
4964
4965
4966                                     ;;****************************************************************************
4967                                     ;* PROGRAM EVENT FLAG DEFINITIONS
4968                                     ;;****************************************************************************
4969
4970
4971
```

```
                                .SBTTL  GLOBAL DATA SECTION

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

                                ;;**************************************************************************
                                ;* STORAGE FOR DEVICE REGISTERS
                                ;;**************************************************************************
002414                                  DESCRIPT        <M8207 DIAG. #1 OF 2>
002414                          L$DESC::
002414  034115  030062  020067            .ASCIZ  /M8207 DIAG. #1 OF 2/
002422  044504  043501  020056
002430  030443  047440  020106
002436  000062
                                        .EVEN

                                ;**************************************************************************
                                ;* PROGRAM CONTROL PARAMETERS
                                ;**************************************************************************
002440  000000                  NEXT:   .WORD   0               ;ADDRESS OF NEXT TEST TO BE EXECUTED
002442  000000                  LOCK:   .WORD   0               ;ADDRESS FOR LOCK CURRENT DATA

                                ;**************************************************************************
                                ;* BUFFERS FOR INPUT-OUTPUT
                                ;**************************************************************************
002444  000000                  TEMP:   0
        002506                  .=.+40
002506  000000                  MDATA:  0
        002550                  .=.+40

                                ;**************************************************************************
                                ;* MISCELLANEOUS STORAGE
                                ;**************************************************************************
002550  000000                  $TMPO:  .WORD   0               ;SCRATCH STORAGE
002552  000000                  LOGDEV: .WORD   0               ;LOGICAL DEVICE NUMBER
002554  000000                  PSTACK: .WORD   0               ;BASE LEVEL PROGRAM STACK POINTER
002556  000000                  SUBRPC: .WORD   0               ;PC OF SUBR CALL FOR ERROR REPORTS
002560  000000                  ERRFLG: .WORD   0               ;SUBROUTINE ERROR FLAG
002562  000000                  RETADR: .WORD   0               ;SUBR ERROR RETURN ADDRESS
002564  000000                  STRTSW: .WORD   0               ;SWITCHES AT START OF PROGRAM
002566  000000                  STAT:   .WORD   0               ;M8200,4,7 STATUS WORD STORAGE
002570  000000                  CLKX:   .WORD   0               ;
002572  000000                  MASKX:  .WORD   0               ;
002574  000000                  SAVSP:  .WORD   0               ;STACK POINTER STORAGE
002576  000000                  SAVPC:  .WORD   0               ;PROGRAM COUNTER STORAGE
002600  000000                  ZERO:   .WORD   0               ;
002602  000001                  ONE:    .WORD   1               ;
002604  037776                  MEMLIM: .WORD   MEMEND          ;HIGHEST LOCATION FOR NFR'S
002606                          MEMSZ:
002606  000000                          .WORD   0
002610  000001                  KMACTV: .BLKW   1               ;M8200,4,7 SELECTED ACTIVE
002612  000001                  KMNUM:  .BLKW   1               ;OCTAL NUMBER OF M8200,4,7
002614  000001                  SAVACT: .BLKW   1               ;ORIGINAL ACTIVE DEVICES
002616  000001                  SAVNUM: .BLKW   1               ;WORKABLE NUMBER
```

```
5023  002620  000000                  FLAG:   .WORD   0               ;SCRATCH STORAGE
5024  002622  000000                  RUN:    .WORD   0               ;POINTER TO RUNNING DEVICES
5025  002624  000000                  MR0:    .WORD   0
5026  002626  000000                  WTYPE:  .WORD   0
5027  002630  000000                  TYPE:   .WORD   0
5028  002632  000000                  $GDADR: .WORD   0               ;CONTAINS ADDRESS OF 'GOOD' DATA
5029  002634  000000                  $BDADR: .WORD   0               ;CONTAINS ADDRESS OF 'BAD' DATA
5030  002636  000000                  $GDDAT: .WORD   0               ;CONTAINS 'GOOD' DATA
5031  002640  000000                  $BDDAT: .WORD   0               ;CONTAINS 'BAD' DATA
5032  002642  000000                          .WORD   0               ;RESERVED--NOT TO BE USED
5033  002644  000000                          .WORD   0
5034  002646  000000                  FTIME:  .WORD   0
5035  002650  000000                  SAVE4:  .WORD   0
5036  002652  000000                  SAVE6:  .WORD   0
5037
5038                                  ;*****************************************************************************
5039                                  ;* DATA PATTERNS
5040                                  ;*****************************************************************************
5041  002654      000     377     000 MEMDAT: .BYTE   0,-1,0,-1,125,252,125,252
      002657      377     125     252
      002662      125     252
5042  002664      000     000     377 SPDAT:  .BYTE   0,0,-1,-1,125,125,252,252
      002667      377     125     125
      002672      252     252
5043                                        .EVEN
5044
5045                                  ;*****************************************************************************
5046                                  ;* PROGRAM CONTROL FLAGS
5047                                  ;*****************************************************************************
5048  002674      000                 INIFLG: .BYTE   0               ;PROGRAM INITIALIZING FLAG
5049          002676                          .EVEN
5050  002676      000                 LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
5051  002677      000                 QV.FLG: .BYTE   0               ;QUICK VERIFY FLAG
5052                                        .EVEN
5053
5054                                  ;*****************************************************************************
5055                                  ;* DEFINITION OF M8200,4,7 STATUS WORDS - STAT1,STAT2,STAT3
5056                                  ;*
5057                                  ;*
5058                                  ;*      STAT1 - BITS 00-08 IS M8200,4,7 VECTOR ADDRESS
5059                                  ;*              BIT15=1 LINE UNIT IS AN M8203
5060                                  ;*              BIT14=0 NO TEST CONNECTOR(S) USED
5061                                  ;*              BIT14=1 H-XXX TEST CONNECTOR WILL BE USED
5062                                  ;*              BIT13=0 LINE UNIT IS AN M8201
5063                                  ;*              BIT13=1 LINE UNIT IS AN M8202
5064                                  ;*              BIT12=1 NO LINE UNIT
5065                                  ;*              BITS 09-11 IS M8200,4,7 PRIORITY LEVEL
5066                                  ;*
5067                                  ;*      STAT2 - LOW BYTE IS SWITCH PACK #1 (DDCMP LINE NUMBER)
5068                                  ;*              HIGH BYTE IS SWITCH PACK #2 (BM873 BOOT ADDRESS)
5069                                  ;*
5070                                  ;*      STAT3 - BIT0=1 DO FREE RUNNING TESTS ON M8200,4,7
5071                                  ;*
5072                                  ;*****************************************************************************
5073  002700  000000                  STAT1:  .WORD   0
5074  002702  000000                  STAT2:  .WORD   0
```

```
5075  002704  000000                  STAT3:  .WORD   0
5076
5077                                  ;*************************************************************************
5078                                  ;* POINTERS TO M8200,4,7 VECTORS AND REGISTERS
5079                                  ;*************************************************************************
5080  002706  000000                  KMRVEC: 0       ;POINTER TO M8200,4,7 RCV INTRPT VECTOR
5081  002710  000000                  KMRLVL: 0       ;POINTER TO M8200,4,7 RCV INTRPT SERVICE PS
5082  002712  000000                  KMTVEC: 0       ;POINTER TO M8200,4,7 TX INTRPT VECTOR
5083  002714  000000                  KMTLVL: 0       ;POINTER TO M8200,4,7 TX INTRPT SERVICE PS
5084  002716  000000                  KMCSR:  0       ;POINTER TO M8200,4,7 CONTROL STATUS REGISTER
5085  002720  000000                  KMCSRH: 0       ;POINTER TO M8200,4,7 CONTROL STATUS REGISTER HIGH BYTE
5086  002722  000000                  KMCTL:  0       ;POINTER TO M8200,4,7 CONTROL OUT REGISTER
5087  002724  000000                  KMP04:  0
5088  002726  000000                  KMP06:  0       ;POINTER TO M8200,4,7 PORT REGISTER - SEL6
5089
5090                                  ;;****  PRIMARY REG ADRS STORAGE FOR THIS UNIT  ******
5091                                  ;THESE LOCATIONS WILL BE LOADED FOR THE CURRENT UNIT, IN INIT CODE
5092  002730                          REGADR:
5093
5094                                  ;;****  STACK USED FOR SUBROUTINE LINKAGE  *****
5095  002730  000100                          .BLKW   100
5096  003130                          SSTACK:
5097
5098
5099
5100
5101
5102
5103
```

```
 5105                                                 .SBTTL  GLOBAL TEXT SECTION
 5106
 5107                                                 ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 5108                                                 ;%       THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
 5109                                                 ;%       MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
 5110                                                 ;%       MORE THAN ONE TEST.
 5111                                                 ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 5112
 5113                                                 ;;**************************************************************************
 5114                                                 ;* NAMES OF DEVICES SUPPORTED BY PROGRAM
 5115                                                 ;;**************************************************************************
 5116   003130                                                DEVTYP  <M8200,4,7>
  (4)   003130                                        L$DVTYP::
  (3)   003130  034115  030062  026060                        .ASCIZ  /M8200,4,7/
  (3)   003136  026064  000067
  (2)                                                         .EVEN
 5117
 5118
 5119                                                 ;
 5120                                                 ; FORMAT STATEMENTS USED IN PRINT CALLS
 5121                                                 ;
 5122
 5129
 5130
 5131
 5132
 5133
```

```
                        .SBTTL  GLOBAL SUBROUTINES

                        ;////////////////////////////////////////////////////////////////////////////////
                        ;/      THE GLOBAL SUBROUTINES ARE CALLED BY MORE THAN ONE TEST
                        ;////////////////////////////////////////////////////////////////////////////////



                        ;-------------------------------------------------------------
                        ; MACRO'S NEEDED TO CALL SUBROUTINES
                        ;-------------------------------------------------------------

                        .MACRO  ERROR,XYX,ZZ
                                MOV     R4,$BDDAT
                                .IF     B ZZ
                                MOV     R2,$GDDAT
                                .ENDC
                                MOV     MR0,$BDADR
                                ERRDF   XYX',EM'XYX',ERR'XYX'
                        .ENDM
                        .MACRO  RERROR  XXX
                                MOV     R4,$BDDAT
                                CLRB    $BDDAT+1
                                CLRB    $GDDAT+1
                                MOV     R2,$GDADR
                                ERRDF   XXX',EM'XXX',ERR'XXX'
                        .ENDM
                        .MACRO  BERROR  XXX
                                MOV     R4,$BDDAT
                                MOV     R5,$GDDAT
                                CLRB    $BDDAT+1
                                CLRB    $GDDAT+1
                                ERRDF   XXX',EM'XXX',ERR'XXX'
                        .ENDM
                        .MACRO  ED$CALL XY
                                .LIST
                                ;******************** TEST 'XY' ********************
                                .NLIST
                                .ENDM
                                .MACRO  BADHEAD
                                .RADIX  10
                                ED$CALL \T$TESTNUM+1
                                .RADIX  8
                                .ENDM
                                .MACRO  K4ONLY  ?N2
                                .LIST
                                ;DON'T DO TEST IF M8200 OR M8204
                                .NLIST
                                CMP     MEMSZ,#2000
                                BNE     N2
                                EXIT    TST
                        N2:
                                .ENDM
                        .MACRO  MYINT
                                .LIST
                                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
```

```
                                        .NLIST
                                .ENDM

                                .MACRO  ROMCLK
                                        .LIST
                                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
                                        .NLIST
                                .ENDM

                                .MACRO  MSTCLR
                                        .LIST
                                        JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
                                        .NLIST
                                .ENDM

003142                          .MSTCLR:
003142  112777  000100  177550          MOVB    #BIT6,@KMCSRH   ;SET INST.
003150  142777  000300  177542          BICB    #BIT6!BIT7,@KMCSRH
003156  000205                          RTS     R5

003160  000024                          .BLKW   20.             ;PATCH AREA.


003230                          ENDBUG:
                                ;       UNSAFE TO PATCH ANY OTHER AREA.


003230                          .ROMCLK:
003230  152777  000002  177462          BISB    #BIT1,@KMCSRH
003236  012577  177464                  MOV     (R5)+,@KMP06
003242  152777  000003  177450          BISB    #BIT1.BIT0,@KMCSRH
003250  142777  000007  177442          BICB    #BIT2!BIT1!BIT0,@KMCSRH
003256  000205                          RTS     R5

003260                          CLRALL:
                                        ;CLEARS C & Z BITS AND BR
003260                                  ROMCLK
 (1)    003260  004537  003230          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
003264  000400                          400                     ;0 TO BR
003266                                  ROMCLK
 (1)    003266  004537  003230          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
003272  063220                          63220                   ;SP(0) TO BR
003274                                  ROMCLK
 (1)    003274  004537  003230          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
003300  060400                          60400                   ;BR,SP(0) + BR
003302  000207                          RTS     PC

003304                          SETBR0:
                                        ;SETS BR0 BIT
003304                                  ROMCLK
 (1)    003304  004537  003230          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
003310  000401                          401                     ;1 TO BR
003312  000207                          RTS     PC
```

```
5243
5244  003314                          SETBR1:
5245                                          ;THIS SUBROJTINE SETS BR1 BIT
5246
5247  003314                          ROMCLK                  ;NEXT WORD IS INSTRUCTION
 (1)  003314  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5248  003320  000402                          000402                  ;BR_002
5249  003322  000207                          RTS     PC
5250
5251  003324                          SETBR4:
5252                                          ;THIS SUBROUTINE SETS BR4 BIT
5253
5254  003324                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION
 (1   003324  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5255  003330  000402                          402
5256  003332  000207                          RTS     PC
5257
5258  003334                          SETBR7:
5259                                          ;THIS SUBROUTINE SETS BR7 BIT
5260
5261  003334                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION
 (1)  003334  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5262  003340  000600                          600
5263  003342  000207                          RTS     PC
5264
5265
5266  003344                          SETZ:
5267                                          ;THIS SUBROUTINE SETS THE Z BIT
5268
5269  003344                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION
 (1)  003344  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5270  003350  000777                          000777                  ;BR_377
5271  003352  000207                          RTS     PC
5272
5273  003354                  RAMDAT:
5274                                          ;THIS SUBROUTINE LOADS R4 WITH THE LOWEST
5275                                          ;8 BITS OF THE CRAM PC.
5276
5277  003354  017605  000000                  MOV     @(SP),R5        ;GOOD DATA
5278  003360  062716  000002                  ADD     #2,(SP)         ;ADJUST STACK
5279  003364  005011                          CLR     (R1)            ;CLEAR BIT10
5280  003366  052711  000400                  BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION IN CRAM THAT
5281                                                                  ;JUMPED TO, IT LOADS BR WITH IT
5282  003372  005011                          CLR     (R1)            ;CLR BIT8
5283  003374                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION
 (1)  003374  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5284  003400  061225                          061225                  ;MOV BR TO PORT 5
5285  003402  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" IN R4
5286  003406  000207                          RTS     PC              ;RETURN
5287
5288
5289  003410                  MEMSET:
5290                                          ;THIS SUBROUTINE LOADS CRAM WITH SPECIAL INSTRUCTIONS
5291                                          ;FOR THE CRAM JUMP TEST. ALL CRAM LOCATIONS ARE LOADED
5292                                          ;WITH INSTRUCTIONS THAT MOVE A 37 TO THE BR, EXCEPT THE
5293                                          ;FOLLOWING CRAM ADDRESSES: 0,1,4,7,525,1777. THESE LOCATIONS
```

```
                                        ;CONTAIN INSTRUCTIONS WHICH LOAD THE BR WITH THE LOWEST
                                        ;8 BITS OF THAT CRAM ADDRESS.

003410  005000                  CLR     R0              ;R0 = CRAM ADDRESS
003412  012711  002000          1$:     MOV     #BIT10,(R1)     ;SET ROMO
003416  010061  000004                  MOV     R0,4(R1)        ;LOAD CRAM ADDRESS
003422  012761  000437  000006          MOV     #437,6(R1)      ;LOAD INSTRUCTION
003430  052711  020000                  BIS     #BIT13,(R1)     ;WRITE INSTRUCTION IN CRAM
003434  005200                  INC     R0              ;NEXT ADDRESS
003436  022700  002000          CMP     #2000,R0        ;DONE YET?
003442  001363                  BNE     1$              ;BR IF NO
003444  005000                  CLR     R0              ;INDEX REGISTER
003446  012711  002000          2$:     MOV     #BIT10,(R1)     ;SET ROMO
003452  016061  003506  000004          MOV     CRAMA(R0),4(R1) ;LOAD CRAM ADDRESS IN SEL4
003460  016061  003522  000006          MOV     INSTU(R0),6(R1) ;LOAD INSTRUCTION TO BE WRITTEN
003466  052711  020000                  BIS     #BIT13,(R1)     ;WRITE CRAM!
003472  005720                  TST     (R0)+           ;NEXT
003474  022700  000014          CMP     #14,R0          ;DONE YET?
003500  001362                  BNE     2$              ;BR IF NO
003502  005011                  CLR     (R1)            ;CLEAR ALL BITS
003504  000207                  RTS     PC              ;RETURN

003506  000000  000001  000004  CRAMA:  .WORD   0,1,4,7,1777,525
003514  000007  001777  000525

003522  000400          INSTU:  000400          ;BR_0
003524  000401                  000401          ;BR_1
003526  000404                  000404          ;BR_4
003530  000407                  000407          ;BR_7
003532  000777                  000777          ;BR_377
003534  000525                  000525          ;BR_125

003536                  SETVEC:
                                        ;THIS SUBROUTINE LOADS THE VECTORS AND VECTOR LEVELS

003536  012577  177144                  MOV     (R5)+,@KMRVEC   ;LOAD BASE VECTOR
003542  012577  177144                  MOV     (R5)+,@KMTVEC   ;LOAD VECTOR + 2
003546  012577  177136                  MOV     (R5)+,@KMRLVL   ;LOAD VECTOR + 4
003552  012577  177136                  MOV     (R5)+,@KMTLVL   ;LOAD VECTOR + 6
003556  000205                  RTS     R5              ;RETURN

003560                  NPRSET:
                                        ;THIS SUBROUTINE LOADS IBUS REGISTERS 0-7
                                        ;WITH NPR INFORMATION (INBA, OUTBA, OUT DATA)

003560  010246                  MOV     R2,-(SP)        ;SAVE R2
003562  005002                  CLR     R2              ;START AT IBUS REG 0
003564  112561  000004          1$:     MOVB    (R5)+,4(R1)     ;LOAD PORT4
003570  042737  000017  003606          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
003576  050237  003606                  BIS     R2,2$           ;ADD ADDRESS TO INSTRUCTION
003602                                  ROMCLK
(1) 003602  004537  003230              JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
003606  122100          2$:     122100                  ;MOVE PORT4 TO IBUS REG
003610  005202                  INC     R2              ;NEXT ADDRESS
003612  022702  000010          CMP     #10,R2          ;ALL DONE?
```

```
 5348  003616  001362                    BNE     1$              ;BR IF NO
 5349  003620  012602                    MOV     (SP)+,R2        ;RESTORE R2
 5350  003622  000205                    RTS     R5              ;RETURN
 5351
 5352
 5353  003624                    MEMLD:
 5354                                    ;THIS SUBROUTINE LOADS THE FIRST 8 LOCATIONS OF MAIN
 5355                                    ;MEMORY WITH THIS DATA: 0,-1,,0,-1,125,252,125,252
 5356
 5357  003624  013637  002550            MOV     @(SP)+,$TMP0    ;PUT POINTER TO DATA IN R0
 5358  003630  062746  000002            ADD     #2,-(SP)        ;ADJUST STACK
 5359
 5360  003634  013700  002550    MEMLD2: MOV     $TMP0,R0        ;GET ADDR.
 5361  003640  012704  000010            MOV     #10,R4          ;DO 8 LOADS
 5362  003644                            ROMCLK
  (1)  003644  004537  003230            JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 5363  003650  010000                    010000                  ;MAR < 0
 5364  003652                            ROMCLK                  ;CLR    MAR HI
  (1)  003652  004537  003230            JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 5365  003656  004000                    004000
 5366  003660  112077  177040    1$:     MOVB    (R0)+,@KMP04    ;LOAD PORT4
 5367  003664                            ROMCLK
  (1)  003664  004537  003230            JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 5368  003670  136500                    136500                  ;MOV DATA TO MEM, AUTO INC MAR
 5369  003672  005304                    DEC     R4              ;DECREMENT COUNT
 5370  003674  001371                    BNE     1$              ;BR IF NOT DONE
 5371
 5372  003676                            ROMCLK                  ;LOAD MEM ADDR. 0
  (1)  003676  004537  003230            JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 5373  003702  010000                    10000
 5374  003704  012703  000010            MOV     #10,R3          ;CHECK 8. MEM LOCS.
 5375  003710  013700  002550            MOV     $TMP0,R0
 5376  003714                    2$:     ROMCLK                  ;READ FROM MEM,PUT INTO PORT 4
  (1)  003714  004537  003230            JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 5377  003720  055224                    55224
 5378
 5379  003722  112037  002636            MOVB    (R0)+,$GDDAT    ;EXPECTED.
 5380  003726  117704  176772            MOVB    @KMP04,R4       ;RECIEVED.
 5381  003732  123704  002636            CMPB    $GDDAT,R4       ;OK?
 5382  003736  001414                    BEQ     3$
 5383  003740                            ERROR   36
  (5)  003756  104455                    TRAP    C$ERDF
  (6)  003760  000044                    .WORD   36
  (6)  003762  005624                    .WORD   EM36
  (6)  003764  010416                    .WORD   ERR36
 5384  003766  000402                    BR      4$
 5385  003770  005303            3$:     DEC     R3              ;CHECKED ALL?
 5386  003772  001350                    BNE     2$              ;NO-DO NEXT ONE.
 5387  003774                    4$:
 5388  003774  000207                    RTS     PC              ;RETURN
 5389
 5390
 5393  003776                    SPLD:
 5394                                    ;THIS SUBROUTINE LOADS THE FIRST 8 SCRATCH PAD
 5395                                    ;LOCATIONS WITH: 0,0,-1,-1,125,125,252,252
 5396
```

```
 5397  003776  013600                        MOV     @(SP)+,R0       ;PUT POINTER TO DATA IN R5
 5398  004000  062746  000002                ADD     #2,-(SP)        ;ADJUST STACK
 5399  004004  005004                        CLR     R4              ;START AT SP ADDRESS 0
 5400  004006  112077  176712        1$:     MOVB    (R0)+,@KMP04    ;LOAD PORT4 WITH DATA
 5401  004012  042737  000017  004030        BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 5402  004020  050437  004030                BIS     R4,2$           ;ADD ADDRESS TO INSTRUCTION
 5403  004024                                ROMCLK
  (1)  004024  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5404  004030  123100                2$:     123100                  ;MOVE DATA TO SP
 5405  004032  005204                        INC     R4              ;INCREMENT COUNT
 5406  004034  022704  000010                CMP     #10,R4          ;DONE YET?
 5407  004040  001362                        BNE     1$              ;BR IF NO
 5408  004042  000207                        RTS     PC              ;RETURN
 5409
 5410
 5411  004044                        CLRC:
 5412                                        ;THIS SUBROUTINE CLEARS THE MICRO PROCESSOR C BIT
 5413
 5414  004044                                ROMCLK
  (1)  004044  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5415  004050  010000                        010000                  ;MAR_0
 5416  004052                                ROMCLK
  (1)  004052  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5417  004056  040400                        040400!<0*20>           ;CLEAR C BIT
 5418  004060  000207                        RTS     PC              ;RETURN
 5419
 5420
 5421  004062                        SETC:
 5422                                        ;THIS SUBROUTINE SETS THE MICRO PROCESSOR C BIT
 5423
 5424  004062                                ROMCLK
  (1)  004062  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5425  004066  010003                        010003                  ;MAR_3
 5426  004070                                ROMCLK
  (1)  004070  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5427  004074  040403                        040403!<0*20>           ;SET C BIT
 5428  004076  000207                        RTS     PC              ;RETURN
 5429
 5430
 5431
 5432
```

```
5434                            .SBTTL  GLOBAL ERROR REPORT SECTION
5435
5436                            ;////////////////////////////////////////////////////////////////////
5437                            ;/      THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
5438                            ;/      THAT ARE USED IN MORE THAN ONE TEST.
5439                            ;////////////////////////////////////////////////////////////////////
```

```
 5441
 5442
 5443
 5444  004100  047045  052045  047045  FM1:    .ASCIZ  /%N%T%N/
       004106    000
 5445  004107     045  031517  051445  TFM1:   .ASCIZ  /%03%S5%03%S5%03%N2/
       004114  022465  031517  051445
       004122  022465  031517  047045
       004130  000062
 5446  004132  047445  022466  031123  TFM2:   .ASCIZ  /%06%S2%06%N2/
       004140  047445  022466  031116
       004146    000
 5447  004147     045  031517  051445  TFM5:   .ASCIZ  /%03%S5%03%N2/
       004154  022465  031517  047045
       004162  000062
 5448  004164  047045  047445  022463  TFM27:  .ASCIZ  /%N%03%S5%03%S7%03%N2/
       004172  032523  047445  022463
       004200  033523  047445  022463
       004206  031116     000
 5449  004211     045  022516  043101  TFM37:  .ASCIZ  /%N%AFAILING ADDRESS IS: %06/
       004216  044501  044514  043516
       004224  040440  042104  042522
       004232  051523  044440  035123
       004240  022440  033117     000
 5450
 5451
```

```
 5453
 5454
 5455  004245     122  043505  051511  EM1:    .ASCIZ  &REGISTER ADDRESS TEST&
       004252  042524  020122  042101
       004260  051104  051505  020123
       004266  042524  052123     000
 5456  004273     111  052502  025123  EM2:    .ASCIZ  &IBUS* REGISTER DUAL ADDRESSING TFST&
       004300  051040  043505  051511
       004306  042524  020122  052504
       004314  046101  040440  042104
       004322  042522  051523  047111
       004330  020107  042524  052123
       004336     000
 5457  004337     111  052502  020123  EM30:   .ASCIZ  ''IBUS REGISTER DUAL ADDRESSING TEST''
       004344  042522  044507  052123
       004352  051105  042040  040525
       004360  020114  042101  051104
       004366  051505  044523  043516
       004374  052040  051505  000124
 5458  004402  051102  051040  043505  EM3:    .ASCIZ  /BR REGISTER DATA TEST/
       004410  051511  042524  020122
       004416  040504  040524  052040
       004424  051505  000124
```

```
 5460  004430  041523  040522  041524  EM4:    .ASCIZ  /SCRATCH PAD DATA TEST/
       004436  020110  040520  020104
       004444  040504  040524  052040
       004452  051505  000124
```

```
 5462  004456  041523  040522  041524  EM5:    .ASCIZ  /SCRATCH PAD DUAL ADDRESSING TEST/
       004464  020110  040520  020104
       004472  052504  046101  040440
       004500  042104  042522  051523
       004506  047111  020107  042524
       004514  052123     000
```

```
 5464  004517     115  044501  020116  EM6:    .ASCIZ  /MAIN MEMORY DATA TEST/
       004524  042515  047515  054522
       004532  042040  052101  020101
       004540  042524  052123     000
 5465  004545     115  044501  020116  EM7:    .ASCIZ  /MAIN MEMORY DUAL ADDRESSING TEST/
       004552  042515  047515  054522
       004560  042040  040525  020114
       004566  042101  051104  051505
       004574  044523  043516  052040
       004602  051505  000124
```

```
 5467  004606  052501  047524  046440  EM10:   .ASCIZ  /AUTO MARINC FUNCTION TEST/
       004614  051101  047111  020103
       004622  052506  041516  044524
       004630  047117  052040  051505
       004636  000124
 5468  004640  050116  020122  042524  EM11:   .ASCIZ  /NPR TEST/
       004646  052123     000
 5469  004651     115  046125  044524  EM12:   .ASCIZ  /MULTIPLE NPR TEST/
       004656  046120  020105  050116
       004664  020122  042524  052123
       004672     000
 5470  004673     116  047117  042440  EM13:   .ASCIZ  /NON EX MEM FAILED/
       004700  020130  042515  020115
       004706  040506  046111  042105
       004714     000
 5471  004715     120  047522  051107  EM14:   .ASCIZ  /PROGRAM CLOCK TEST/
       004722  046501  041440  047514
       004730  045503  052040  051505
       004736  000124
 5472  004740  046101  020125  052506  EM15:   .ASCIZ  /ALU FUNCTION WITH C BIT CLEAR TEST/
       004746  041516  044524  047117
       004754  053440  052111  020110
       004762  020103  044502  020124
       004770  046103  040505  020122
       004776  042524  052123     000
```

```
 5474  005003     120  053517  051105  EM16:   .ASCIZ  /POWER FAIL: BUS INIT WAS NOT BLOCKED/
       005010  043040  044501  035114
       005016  041040  051525  044440
       005024  044516  020124  040527
       005032  020123  047516  020124
       005040  046102  041517  042513
       005046  000104
 5475  005050                          EM35:
 5476  005050  047506  041522  020105  EM17:   .ASCIZ  /FORCE POWER FAIL ERROR/
       005056  047520  042527  020122
       005064  040506  046111  042440
       005072  051122  051117     000
 5477  005077     116  044517  042523  EM20:   .ASCIZ  /NOISE TEST ON IBUS*,IBUS,SPAD,MEMORY/
       005104  052040  051505  020124
       005112  047117  044440  052502
       005120  025123  044454  052502
       005126  026123  050123  042101
       005134  046454  046505  051117
       005142  000131
 5478  005144  046101  020125  020103  EM21:   .ASCIZ  /ALU C BIT TEST FAILURE/
       005152  044502  020124  042524
       005160  052123  043040  044501
       005166  052514  042522     000
 5479  005173     124  046511  020105  EM22:   .ASCIZ  /TIME OUT ERROR/
       005200  052517  020124  051105
       005206  047522  000122
 5480  005212  046101  020125  052506  EM23:   .ASCIZ  /ALU FUNCTION TEST WITH C BIT SET/
       005220  041516  044524  047117
       005226  052040  051505  020124
       005234  044527  044124  041440
       005242  041040  052111  051440
       005250  052105     000
 5481  005253     125  041520  051440  EM24:   .ASCIZ  /UPC SEQUENCE ERROR/
       005260  050505  042525  041516
       005266  020105  051105  047522
       005274  000122
 5482  005276  050125  043040  044501  EM31:   .ASCIZ  'UP FAILED TO INTERRUPT''
       005304  042514  020104  047524
       005312  044440  052116  051105
       005320  052522  052120     000
 5483  005325     125  020120  047111  EM32:   .ASCIZ  'UP INTERRUPTED TO WRONG VECTOR''
       005332  042524  051122  050125
       005340  042524  020104  047524
       005346  053440  047522  043516
       005354  053040  041505  047524
       005362  000122
 5484  005364  047125  054105  042520  EM33:   .ASCIZ  'UNEXPECTED INTERRUPT FROM UP''
       005372  052103  042105  044440
       005400  052116  051105  052522
       005406  052120  043040  047522
       005414  020115  050125     000
 5485  005421     101  052514  043040  EM34:   .ASCIZ  ''ALU FLAG TEST''
       005426  040514  020107  042524
       005434  052123     000
 5486  005437     110  046105  020114  EM25:   .ASCIZ  /HELL RAISER TEST/
       005444  040522  051511  051105
```

```
                005452  052040  051505  000124
 5487           005460  040515  047111  040524  EM26:   .ASCIZ  /MAINTANCE REGISTER ERROR/
                005466  041516  020105  042522
                005474  044507  052123  051105
                005502  042440  051122  051117
                005510     000
 5488           005511     111  052502  025123  EM27:   .ASCIZ  'IBUS* WRITE/READ ERROR'
                005516  053440  044522  042524
                005524  051057  040505  020104
                005532  051105  047522  000122
 5489           005540  047111  052123  052522  EM28:   .ASCIZ  /INSTRUCTION TEST FAILURE/
                005546  052103  047511  020116
                005554  042524  052123  043040
                005562  044501  052514  042522
                005570     000
 5490           005571     111  052502  027523  EM29:   .ASCIZ  'IBUS/OBUS WRITE/READ ERROR'
                005576  041117  051525  053440
                005604  044522  042524  051057
                005612  040505  020104  051105
                005620  047522  000122
 5491
 5492           005624  047511  020120  040515  EM36:   .ASCIZ  'IOP MAIN MEM. LOAD ERROR-RUN MCPU MEM. DIAG.'
                005632  047111  046440  046505
                005640  020056  047514  042101
                005646  042440  051122  051117
                005654  051055  047125  046440
                005662  050103  020125  042515
                005670  027115  042040  040511
                005676  027107     000
 5493           005701     000                  EM37:   .ASCIZ  //
 5494
 5495
```

5497

```
5499  005702      000                 DH0:    .ASCIZ  //
```

```
 5501  005703     107  047517  020104  DH1:    .ASCIZ  /GOOD    BAD     REGISTER/
       005710  020040  041040  042101
       005716  020040  020040  051040
       005724  043505  051511  042524
       005732  000122
 5502  005734  047507  042117  020040  DH2:    .ASCIZ  /GOOD    BAD/
       005742  020040  040502  000104
 5503  005750  047507  042117  020040  DH3:    .ASCIZ  /GOOD    BAD     ADDRESS/
       005756  020040  040502  020104
       005764  020040  020040  042101
       005772  051104  051505  000123
 5504  006000  047507  042117  020040  DH4:    .ASCIZ  /GOOD    BAD/
       006006  020040  040502  000104
 5505  006014  042522  027107  042440  DH27:   .ASCIZ  /REG. EXPECTED FOUND/
       006022  050130  041505  042524
       006030  020104  047506  047125
       006036  000104
 5506
 5507
 5508
 5509
 5510                                  .EVEN                 1
```

```
5512
5513
5514
```

```
5516                    ;-------------------------------------------------------------------------------
5517                    ; MACRO'S NEEDED TO REPORT ERRORS
```

```
5519                                    ;-----------------------------------------------------------------------
5520                                    .MACRO  MDT0
5521                                    .ENDM
5522
5523                                    .MACRO  MDT1
5524                                            PRINTB #TFM1,$GDDAT,$BDDAT,$GDADR
5525                                    .ENDM
5526
5527                                    .MACRO MDT2
```

```
5529                                    PRINTB #TFM2,$GDDAT,$BDDAT
5530                                .ENDM
5531
5532                                .MACRO  MDT5
5533                                    PRINTB  #TFM5,$GDDAT,$BDDAT
5534                                .ENDM
```

```
5536                                    .MACRO  MDT27
5537                                            PRINTB #TFM27,MRO,$GDDAT,$BDDAT
5538                                    .ENDM
5539
5540                                    .MACRO  $MD,ERNB,ERHM,ERFM
5541                                    .NLIST
5542                                    ; ERNB = ERROR NUMBER
5543                                    ; ERFM = FORMAT NUMBER
5544                                    ; ERHM = HEADER NUMBER
5545                                    .LIST
5546                                            BGNMSG  ERR'ERNB'
5547                                            PRINTB  #FM1,#DH'ERHM'
5548                                            MDT'ERFM'
5549                                            ENDMSG
```

```
 5551                                   .ENDM
 5552
 5553
 5554
 5555
 5556  006040                                  $MD     1,2,2
  (4)  006040                           ERR1::
  (9)  006040  012746  005734                  MOV     #DH2,-(SP)
  (8)  006044  012746  004100                  MOV     #FM1,-(SP)
  (7)  006050  012746  000002                  MOV     #2,-(SP)
  (4)  006054  010600                          MOV     SP,R0
  (5)  006056  104414                          TRAP    C$PNTB
  (5)  006060  062706  000006                  ADD     #6,SP
 (11)  006064  013746  002640                  MOV     $BDDAT,-(SP)
 (10)  006070  013746  002636                  MOV     $GDDAT,-(SP)
  (9)  006074  012746  004132                  MOV     #TFM2,-(SP)
  (8)  006100  012746  000003                  MOV     #3,-(SP)
  (5)  006104  010600                          MOV     SP,R0
  (6)  006106  104414                          TRAP    C$PNTB
  (6)  006110  062706  000010                  ADD     #10,SP
  (4)  006114                           L10003:
  (4)  006114  104423                          TRAP    C$MSG
 5557  006116                                  $MD     2,2,2
  (4)  006116                           ERR2::
  (9)  006116  012746  005734                  MOV     #DH2,-(SP)
  (8)  006122  012746  004100                  MOV     #FM1,-(SP)
  (7)  006126  012746  000002                  MOV     #2,-(SP)
  (4)  006132  010600                          MOV     SP,R0
  (5)  006134  104414                          TRAP    C$PNTB
  (5)  006136  062706  000006                  ADD     #6,SP
 (11)  006142  013746  002640                  MOV     $BDDAT,-(SP)
 (10)  006146  013746  002636                  MOV     $GDDAT,-(SP)
  (9)  006152  012746  004132                  MOV     #TFM2,-(SP)
  (8)  006156  012746  000003                  MOV     #3,-(SP)
  (5)  006162  010600                          MOV     SP,R0
  (6)  006164  104414                          TRAP    C$PNTB
  (6)  006166  062706  000010                  ADD     #10,SP
  (4)  006172                           L10004:
  (4)  006172  104423                          TRAP    C$MSG
 5558  006174                                  $MD     3,2,2
  (4)  006174                           ERR3::
  (9)  006174  012746  005734                  MOV     #DH2,-(SP)
  (8)  006200  012746  004100                  MOV     #FM1,-(SP)
  (7)  006204  012746  000002                  MOV     #2,-(SP)
  (4)  006210  010600                          MOV     SP,R0
  (5)  006212  104414                          TRAP    C$PNTB
  (5)  006214  062706  000006                  ADD     #6,SP
 (11)  006220  013746  002640                  MOV     $BDDAT,-(SP)
 (10)  006224  013746  002636                  MOV     $GDDAT,-(SP)
  (9)  006230  012746  004132                  MOV     #TFM2,-(SP)
  (8)  006234  012746  000003                  MOV     #3,-(SP)
  (5)  006240  010600                          MOV     SP,R0
  (6)  006242  104414                          TRAP    C$PNTB
  (6)  006244  062706  000010                  ADD     #10,SP
  (4)  006250                           L10005:
  (4)  006250  104423                          TRAP    C$MSG
```

```
5559  006252                                  $MD     4,1,1
 (4)  006252                      ERR4::
 (9)  006252  012746  005703              MOV     #DH1,-(SP)
 (8)  006256  012746  004100              MOV     #FM1,-(SP)
 (7)  006262  012746  000002              MOV     #2,-(SP)
 (4)  006266  010600                      MOV     SP,R0
 (5)  006270  104414                      TRAP    C$PNTB
 (5)  006272  062706  000006              ADD     #6,SP
 (12) 006276  013746  002632              MOV     $GDADR,-(SP)
 (11) 006302  013746  002640              MOV     $BDDAT,-(SP)
 (10) 006306  013746  002636              MOV     $GDDAT,-(SP)
 (9)  006312  012746  004107              MOV     #TFM1,-(SP)
 (8)  006316  012746  000004              MOV     #4,-(SP)
 (5)  006322  010600                      MOV     SP,R0
 (6)  006324  104414                      TRAP    C$PNTB
 (6)  006326  062706  000012              ADD     #12,SP
 (4)  006332                      L10006:
 (4)  006332  104423                      TRAP    C$MSG
5560  006334                              $MD     5,1,1
 (4)  006334                      ERR5::
 (9)  006334  012746  005703              MOV     #DH1,-(SP)
 (8)  006340  012746  004100              MOV     #FM1,-(SP)
 (7)  006344  012746  000002              MOV     #2,-(SP)
 (4)  006350  010600                      MOV     SP,R0
 (5)  006352  104414                      TRAP    C$PNTB
 (5)  006354  062706  000006              ADD     #6,SP
 (12) 006360  013746  002632              MOV     $GDADR,-(SP)
 (11) 006364  013746  002640              MOV     $BDDAT,-(SP)
 (10) 006370  013746  002636              MOV     $GDDAT,-(SP)
 (9)  006374  012746  004107              MOV     #TFM1,-(SP)
 (8)  006400  012746  000004              MOV     #4,-(SP)
 (5)  006404  010600                      MOV     SP,R0
 (6)  006406  104414                      TRAP    C$PNTB
 (6)  006410  062706  000012              ADD     #12,SP
 (4)  006414                      L10007:
 (4)  006414  104423                      TRAP    C$MSG
5561  006416                              $MD     6,3,1
 (4)  006416                      ERR6::
 (9)  006416  012746  005750              MOV     #DH3,-(SP)
 (8)  006422  012746  004100              MOV     #FM1,-(SP)
 (7)  006426  012746  000002              MOV     #2,-(SP)
 (4)  006432  010600                      MOV     SP,R0
 (5)  006434  104414                      TRAP    C$PNTB
 (5)  006436  062706  000006              ADD     #6,SP
 (12) 006442  013746  002632              MOV     $GDADR,-(SP)
 (11) 006446  013746  002640              MOV     $BDDAT,-(SP)
 (10) 006452  013746  002636              MOV     $GDDAT,-(SP)
 (9)  006456  012746  004107              MOV     #TFM1,-(SP)
 (8)  006462  012746  000004              MOV     #4,-(SP)
 (5)  006466  010600                      MOV     SP,R0
 (6)  006470  104414                      TRAP    C$PNTB
 (6)  006472  062706  000012              ADD     #12,SP
 (4)  006476                      L10010:
 (4)  006476  104423                      TRAP    C$MSG
5562  006500                              $MD     7,3,1
 (4)  006500                      ERR7::
```

```
   (9)  006500  012746  005750          MOV     #DH3,-(SP)
   (8)  006504  012746  004100          MOV     #FM1,-(SP)
   (7)  006510  012746  000002          MOV     #2,-(SP)
   (4)  006514  010600                  MOV     SP,R0
   (5)  006516  104414                  TRAP    C$PNTB
   (5)  006520  062706  000006          ADD     #6,SP
  (12)  006524  013746  002632          MOV     $GDADR,-(SP)
  (11)  006530  013746  002640          MOV     $BDDAT,-(SP)
  (10)  006534  013746  002636          MOV     $GDDAT,-(SP)
   (9)  006540  012746  004107          MOV     #TFM1,-(SP)
   (8)  006544  012746  000004          MOV     #4,-(SP)
   (5)  006550  010600                  MOV     SP,R0
   (6)  006552  104414                  TRAP    C$PNTB
   (6)  006554  062706  000012          ADD     #12,SP
   (4)  006560                  L10011:
   (4)  006560  104423                  TRAP    C$MSG
 5563   006562                          $MD     10,3,1
   (4)  006562                  ERR10::
   (9)  006562  012746  005750          MOV     #DH3,-(SP)
   (8)  006566  012746  004100          MOV     #FM1,-(SP)
   (7)  006572  012746  000002          MOV     #2,-(SP)
   (4)  006576  010600                  MOV     SP,R0
   (5)  006600  104414                  TRAP    C$PNTB
   (5)  006602  062706  000006          ADD     #6,SP
  (12)  006606  013746  002632          MOV     $GDADR,-(SP)
  (11)  006612  013746  002640          MOV     $BDDAT,-(SP)
  (10)  006616  013746  002636          MOV     $GDDAT,-(SP)
   (9)  006622  012746  004107          MOV     #TFM1,-(SP)
   (8)  006626  012746  000004          MOV     #4,-(SP)
   (5)  006632  010600                  MOV     SP,R0
   (6)  006634  104414                  TRAP    C$PNTB
   (6)  006636  062706  000012          ADD     #12,SP
   (4)  006642                  L10012:
   (4)  006642  104423                  TRAP    C$MSG
 5564   006644                          $MD     11,2,2
   (4)  006644                  ERR11::
   (9)  006644  012746  005734          MOV     #DH2,-(SP)
   (8)  006650  012746  004100          MOV     #FM1,-(SP)
   (7)  006654  012746  000002          MOV     #2,-(SP)
   (4)  006660  010600                  MOV     SP,R0
   (5)  006662  104414                  TRAP    C$PNTB
   (5)  006664  062706  000006          ADD     #6,SP
  (11)  006670  013746  002640          MOV     $BDDAT,-(SP)
  (10)  006674  013746  002636          MOV     $GDDAT,-(SP)
   (9)  006700  012746  004132          MOV     #TFM2,-(SP)
   (8)  006704  012746  000003          MOV     #3,-(SP)
   (5)  006710  010600                  MOV     SP,R0
   (6)  006712  104414                  TRAP    C$PNTB
   (6)  006714  062706  000010          ADD     #10,SP
   (4)  006720                  L10013:
   (4)  006720  104423                  TRAP    C$MSG
 5565   006722                          $MD     12,2,2
   (4)  006722                  ERR12::
   (9)  006722  012746  005734          MOV     #DH2,-(SP)
   (8)  006726  012746  004100          MOV     #FM1,-(SP)
   (7)  006732  012746  000002          MOV     #2,-(SP)
```

```
   (4)  006736  010600                        MOV     SP,R0
   (5)  006740  104414                        TRAP    C$PNTB
   (5)  006742  062706  000006                ADD     #6,SP
  (11)  006746  013746  002640                MOV     $BDDAT,-(SP)
  (10)  006752  013746  002636                MOV     $GDDAT,-(SP)
   (9)  006756  012746  004132                MOV     #TFM2,-(SP)
   (8)  006762  012746  000003                MOV     #3,-(SP)
   (5)  006766  010600                        MOV     SP,R0
   (6)  006770  104414                        TRAP    C$PNTB
   (6)  006772  062706  000010                ADD     #10,SP
   (4)  006776                        L10014:
   (4)  006776  104423                        TRAP    C$MSG
 5566   007000                                $MD     13,0,0
   (4)  007000                        ERR13::
   (9)  007000  012746  005702                MOV     #DH0,-(SP)
   (8)  007004  012746  004100                MOV     #FM1,-(SP)
   (7)  007010  012746  000002                MOV     #2,-(SP)
   (4)  007014  010600                        MOV     SP,R0
   (5)  007016  104414                        TRAP    C$PNTB
   (5)  007020  062706  000006                ADD     #6,SP
   (4)  007024                        L10015:
   (4)  007024  104423                        TRAP    C$MSG
```

```
5568  007026                                  $MD     14,2,2
 (4)  007026                        ERR14::
 (9)  007026  012746  005734                  MOV     #DH2,-(SP)
 (8)  007032  012746  004100                  MOV     #FM1,-(SP)
 (7)  007036  012746  000002                  MOV     #2,-(SP)
 (4)  007042  010600                          MOV     SP,R0
 (5)  007044  104414                          TRAP    C$PNTB
 (5)  007046  062706  000006                  ADD     #6,SP
 (11) 007052  013746  002640                  MOV     $BDDAT,-(SP)
 (10) 007056  013746  002636                  MOV     $GDDAT,-(SP)
 (9)  007062  012746  004132                  MOV     #TFM2,-(SP)
 (8)  007066  012746  000003                  MOV     #3,-(SP)
 (5)  007072  010600                          MOV     SP,R0
 (6)  007074  104414                          TRAP    C$PNTB
 (6)  007076  062706  000010                  ADD     #10,SP
 (4)  007102                        L10016:
 (4)  007102  104425                          TRAP    C$MSG
```

```
 5570  007104                              $MD     15,4,5
  (4)  007104                      ERR15::
  (9)  007104  012746  006000              MOV     #DH4,-(SP)
  (8)  007110  012746  004100              MOV     #FM1,-(SP)
  (7)  007114  012746  000002              MOV     #2,-(SP)
  (4)  007120  010600                      MOV     SP,R0
  (5)  007122  104414                      TRAP    C$PNTB
  (5)  007124  062706  000006              ADD     #6,SP
 (11)  007130  013746  002640              MOV     $BDDAT,-(SP)
 (10)  007134  013746  002636              MOV     $GDDAT,-(SP)
  (9)  007140  012746  004147              MOV     #TFM5,-(SP)
  (8)  007144  012746  000003              MOV     #3,-(SP)
  (5)  007150  010600                      MOV     SP,R0
  (6)  007152  104414                      TRAP    C$PNTB
  (6)  007154  062706  000010              ADD     #10,SP
  (4)  007160                      L10017:
  (4)  007160  104423                      TRAP    C$MSG
 5571  007162                              $MD     16,0,0
  (4)  007162                      ERR16::
  (9)  007162  012746  005702              MOV     #DH0,-(SP)
  (8)  007166  012746  004100              MOV     #FM1,-(SP)
  (7)  007172  012746  000002              MOV     #2,-(SP)
  (4)  007176  010600                      MOV     SP,R0
  (5)  007200  104414                      TRAP    C$PNTB
  (5)  007202  062706  000006              ADD     #6,SP
  (4)  007206                      L10020:
  (4)  007206  104423                      TRAP    C$MSG
 5572  007210                              $MD     17,0,0
  (4)  007210                      ERR17::
  (9)  007210  012746  005702              MOV     #DH0,-(SP)
  (8)  007214  012746  004100              MOV     #FM1,-(SP)
  (7)  007220  012746  000002              MOV     #2,-(SP)
  (4)  007224  010600                      MOV     SP,R0
  (5)  007226  104414                      TRAP    C$PNTB
  (5)  007230  062706  000006              ADD     #6,SP
  (4)  007234                      L10021:
  (4)  007234  104423                      TRAP    C$MSG
 5573  007236                              $MD     20,2,2
  (4)  007236                      ERR20::
  (9)  007236  012746  005734              MOV     #DH2,-(SP)
  (8)  007242  012746  004100              MOV     #FM1,-(SP)
  (7)  007246  012746  000002              MOV     #2,-(SP)
  (4)  007252  010600                      MOV     SP,R0
  (5)  007254  104414                      TRAP    C$PNTB
  (5)  007256  062706  000006              ADD     #6,SP
 (11)  007262  013746  002640              MOV     $BDDAT,-(SP)
 (10)  007266  013746  002636              MOV     $GDDAT,-(SP)
  (9)  007272  012746  004132              MOV     #TFM2,-(SP)
  (8)  007276  012746  000003              MOV     #3,-(SP)
  (5)  007302  010600                      MOV     SP,R0
  (6)  007304  104414                      TRAP    C$PNTB
  (6)  007306  062706  000010              ADD     #10,SP
  (4)  007312                      L10022:
  (4)  007312  104423                      TRAP    C$MSG
 5574  007314                              $MD     21,0,0
  (4)  007314                      ERR21::
```

```
   (9)  007314  012746  005702          MOV     #DH0,-(SP)
   (8)  007320  012746  004100          MOV     #FM1,-(SP)
   (7)  007324  012746  000002          MOV     #2,-(SP)
   (4)  007330  010600                  MOV     SP,R0
   (5)  007332  104414                  TRAP    C$PNTB
   (5)  007334  062706  000006          ADD     #6,SP
   (4)  007340                  L10023:
   (4)  007340  104423                  TRAP    C$MSG
 5575   007342                          $MD     22,0,0
   (4)  007342                  ERR22::
   (9)  007342  012746  005702          MOV     #DH0,-(SP)
   (8)  007346  012746  004100          MOV     #FM1,-(SP)
   (7)  007352  012746  000002          MOV     #2,-(SP)
   (4)  007356  010600                  MOV     SP,R0
   (5)  007360  104414                  TRAP    C$PNTB
   (5)  007362  062706  000006          ADD     #6,SP
   (4)  007366                  L10024:
   (4)  007366  104423                  TRAP    C$MSG
 5576   007370                          $MD     23,4,5
   (4)  007370                  ERR23::
   (9)  007370  012746  006000          MOV     #DH4,-(SP)
   (8)  007374  012746  004100          MOV     #FM1,-(SP)
   (7)  007400  012746  000002          MOV     #2,-(SP)
   (4)  007404  010600                  MOV     SP,R0
   (5)  007406  104414                  TRAP    C$PNTB
   (5)  007410  062706  000006          ADD     #6,SP
  (11)  007414  013746  002640          MOV     $BDDAT,-(SP)
  (10)  007420  013746  002636          MOV     $GDDAT,-(SP)
   (9)  007424  012746  004147          MOV     #TFM5,-(SP)
   (8)  007430  012746  000003          MOV     #3,-(SP)
   (5)  007434  010600                  MOV     SP,R0
   (6)  007436  104414                  TRAP    C$PNTB
   (6)  007440  062706  000010          ADD     #10,SP
   (4)  007444                  L10025:
   (4)  007444  104423                  TRAP    C$MSG
 5577   007446                          $MD     24,0,0
   (4)  007446                  ERR24::
   (9)  007446  012746  005702          MOV     #DH0,-(SP)
   (8)  007452  012746  004100          MOV     #FM1,-(SP)
   (7)  007456  012746  000002          MOV     #2,-(SP)
   (4)  007462  010600                  MOV     SP,R0
   (5)  007464  104414                  TRAP    C$PNTB
   (5)  007466  062706  000006          ADD     #6,SP
   (4)  007472                  L10026:
   (4)  007472  104423                  TRAP    C$MSG
 5578   007474                          $MD     25,2,2
   (4)  007474                  ERR25::
   (9)  007474  012746  005734          MOV     #DH2,-(SP)
   (8)  007500  012746  004100          MOV     #FM1,-(SP)
   (7)  007504  012746  000002          MOV     #2,-(SP)
   (4)  007510  010600                  MOV     SP,R0
   (5)  007512  104414                  TRAP    C$PNTB
   (5)  007514  062706  000006          ADD     #6,SP
  (11)  007520  013746  002640          MOV     $BDDAT,-(SP)
  (10)  007524  013746  002636          MOV     $GDDAT,-(SP)
   (9)  007530  012746  004132          MOV     #TFM2,-(SP)
```

```
   (8)  007534  012746  000003          MOV     #3,-(SP)
   (5)  007540  010600                  MOV     SP,R0
   (6)  007542  104414                  TRAP    C$PNTB
   (6)  007544  062706  000010          ADD     #10,SP
   (4)  007550                  L10027:
   (4)  007550  104423                  TRAP    C$MSG
 5579   007552                          $MD     26,2,2
   (4)  007552                  ERR26::
   (9)  007552  012746  005734          MOV     #DH2,-(SP)
   (8)  007556  012746  004100          MOV     #FM1,-(SP)
   (7)  007562  012746  000002          MOV     #2,-(SP)
   (4)  007566  010600                  MOV     SP,R0
   (5)  007570  104414                  TRAP    C$PNTB
   (5)  007572  062706  000006          ADD     #6,SP
  (11)  007576  013746  002640          MOV     $BDDAT,-(SP)
  (10)  007602  013746  002636          MOV     $GDDAT,-(SP)
   (9)  007606  012746  004132          MOV     #TFM2,-(SP)
   (8)  007612  012746  000003          MOV     #3,-(SP)
   (5)  007616  010600                  MOV     SP,R0
   (6)  007620  104414                  TRAP    C$PNTB
   (6)  007622  062706  000010          ADD     #10,SP
   (4)  007626                  L10030:
   (4)  007626  104423                  TRAP    C$MSG
 5580   007630                          $MD     27,27,27
   (4)  007630                  ERR27::
   (9)  007630  012746  006014          MOV     #DH27,-(SP)
   (8)  007634  012746  004100          MOV     #FM1,-(SP)
   (7)  007640  012746  000002          MOV     #2,-(SP)
   (4)  007644  010600                  MOV     SP,R0
   (5)  007646  104414                  TRAP    C$PNTB
   (5)  007650  062706  000006          ADD     #6,SP
  (12)  007654  013746  002640          MOV     $BDDAT,-(SP)
  (11)  007660  013746  002636          MOV     $GDDAT,-(SP)
  (10)  007664  013746  002624          MOV     MR0,-(SP)
   (9)  007670  012746  004164          MOV     #TFM27,-(SP)
   (8)  007674  012746  000004          MOV     #4,-(SP)
   (5)  007700  010600                  MOV     SP,R0
   (6)  007702  104414                  TRAP    C$PNTB
   (6)  007704  062706  000012          ADD     #12,SP
   (4)  007710                  L10031:
   (4)  007710  104423                  TRAP    C$MSG
 5581   007712                          $MD     28,2,2
   (4)  007712                  ERR28::
   (9)  007712  012746  005734          MOV     #DH2,-(SP)
   (8)  007716  012746  004100          MOV     #FM1,-(SP)
   (7)  007722  012746  000002          MOV     #2,-(SP)
   (4)  007726  010600                  MOV     SP,R0
   (5)  007730  104414                  TRAP    C$PNTB
   (5)  007732  062706  000006          ADD     #6,SP
  (11)  007736  013746  002640          MOV     $BDDAT,-(SP)
  (10)  007742  013746  002636          MOV     $GDDAT,-(SP)
   (9)  007746  012746  004132          MOV     #TFM2,-(SP)
   (8)  007752  012746  000003          MOV     #3,-(SP)
   (5)  007756  010600                  MOV     SP,R0
   (6)  007760  104414                  TRAP    C$PNTB
   (6)  007762  062706  000010          ADD     #10,SP
```

```
   (4)  007766
   (4)  007766  104423
 5582   007770
   (4)  007770
   (9)  007770  012746  006014
   (8)  007774  012746  004100
   (7)  010000  012746  000002
   (4)  010004  010600
   (5)  010006  104414
   (5)  010010  062706  000006
  (12)  010014  013746  002640
  (11)  010020  013746  002636
  (10)  010024  013746  002624
   (9)  010030  012746  004164
   (8)  010034  012746  000004
   (5)  010040  010600
   (6)  010042  104414
   (6)  010044  062706  000012
   (4)  010050
   (4)  010050  104423
 5583   010052
   (4)  010052
   (9)  010052  012746  005734
   (8)  010056  012746  004100
   (7)  010062  012746  000002
   (4)  010066  010600
   (5)  010070  104414
   (5)  010072  062706  000006
  (11)  010076  013746  002640
  (10)  010102  013746  002636
   (9)  010106  012746  004132
   (8)  010112  012746  000003
   (5)  010116  010600
   (6)  010120  104414
   (6)  010122  062706  000010
   (4)  010126
   (4)  010126  104423
 5584   010130
   (4)  010130
   (9)  010130  012746  005702
   (8)  010134  012746  004100
   (7)  010140  012746  000002
   (4)  010144  010600
   (5)  010146  104414
   (5)  010150  062706  000006
   (4)  010154
   (4)  010154  104423
 5585   010156
   (4)  010156
   (9)  010156  012746  005702
   (8)  010162  012746  004100
   (7)  010166  012746  000002
   (4)  010172  010600
   (5)  010174  104414
   (5)  010176  062706  000006
   (4)  010202
```

```
L10032:
        TRAP    C$MSG
        $MD     29,27,27
ERR29::
        MOV     #DH27,-(SP)
        MOV     #FM1,-(SP)
        MOV     #2,-(SP)
        MOV     SP,R0
        TRAP    C$PNTB
        ADD     #6,SP
        MOV     $BDDAT,-(SP)
        MOV     $GDDAT,-(SP)
        MOV     MR0,-(SP)
        MOV     #TFM27,-(SP)
        MOV     #4,-(SP)
        MOV     SP,R0
        TRAP    C$PNTB
        ADD     #12,SP
L10033:
        TRAP    C$MSG
        $MD     30,2,2
ERR30::
        MOV     #DH2,-(SP)
        MOV     #FM1,-(SP)
        MOV     #2,-(SP)
        MOV     SP,R0
        TRAP    C$PNTB
        ADD     #6,SP
        MOV     $BDDAT,-(SP)
        MOV     $GDDAT,-(SP)
        MOV     #TFM2,-(SP)
        MOV     #3,-(SP)
        MOV     SP,R0
        TRAP    C$PNTB
        ADD     #10,SP
L10034:
        TRAP    C$MSG
        $MD     31,0,0
ERR31::
        MOV     #DH0,-(SP)
        MOV     #FM1,-(SP)
        MOV     #2,-(SP)
        MOV     SP,R0
        TRAP    C$PNTB
        ADD     #6,SP
L10035:
        TRAP    C$MSG
        $MD     32,0,0
ERR32::
        MOV     #DH0,-(SP)
        MOV     #FM1,-(SP)
        MOV     #2,-(SP)
        MOV     SP,R0
        TRAP    C$PNTB
        ADD     #6,SP
L10036:
```

```
 (4)  010202  104423                     TRAP    C$MSG
5586  010204                             $MD     33,2,2
 (4)  010204                    ERR33::
 (9)  010204  012746  005734             MOV     #DH2,-(SP)
 (8)  010210  012746  004100             MOV     #FM1,-(SP)
 (7)  010214  012746  000002             MOV     #2,-(SP)
 (4)  010220  010600                     MOV     SP,R0
 (5)  010222  104414                     TRAP    C$PNTB
 (5)  010224  062706  000006             ADD     #6,SP
(11)  010230  013746  002640             MOV     $BDDAT,-(SP)
(10)  010234  013746  002636             MOV     $GDDAT,-(SP)
 (9)  010240  012746  004132             MOV     #TFM2,-(SP)
 (8)  010244  012746  000003             MOV     #3,-(SP)
 (5)  010250  010600                     MOV     SP,R0
 (6)  010252  104414                     TRAP    C$PNTB
 (6)  010254  062706  000010             ADD     #10,SP
 (4)  010260                    L10037:
 (4)  010260  104423                     TRAP    C$MSG
5587  010262                             $MD     34,2,2
 (4)  010262                    ERR34::
 (9)  010262  012746  005734             MOV     #DH2,-(SP)
 (8)  010266  012746  004100             MOV     #FM1,-(SP)
 (7)  010272  012746  000002             MOV     #2,-(SP)
 (4)  010276  010600                     MOV     SP,R0
 (5)  010300  104414                     TRAP    C$PNTB
 (5)  010302  062706  000006             ADD     #6,SP
(11)  010306  013746  002640             MOV     $BDDAT,-(SP)
(10)  010312  013746  002636             MOV     $GDDAT,-(SP)
 (9)  010316  012746  004132             MOV     #TFM2,-(SP)
 (8)  010322  012746  000003             MOV     #3,-(SP)
 (5)  010326  010600                     MOV     SP,R0
 (6)  010330  104414                     TRAP    C$PNTB
 (6)  010332  062706  000010             ADD     #10,SP
 (4)  010336                    L10040:
 (4)  010336  104423                     TRAP    C$MSG
5588  010340                             $MD     35,2,2
 (4)  010340                    ERR35::
 (9)  010340  012746  005734             MOV     #DH2,-(SP)
 (8)  010344  012746  004100             MOV     #FM1,-(SP)
 (7)  010350  012746  000002             MOV     #2,-(SP)
 (4)  010354  010600                     MOV     SP,R0
 (5)  010356  104414                     TRAP    C$PNTB
 (5)  010360  062706  000006             ADD     #6,SP
(11)  010364  013746  002640             MOV     $BDDAT,-(SP)
(10)  010370  013746  002636             MOV     $GDDAT,-(SP)
 (9)  010374  012746  004132             MOV     #TFM2,-(SP)
 (8)  010400  012746  000003             MOV     #3,-(SP)
 (5)  010404  010600                     MOV     SP,R0
 (6)  010406  104414                     TRAP    C$PNTB
 (6)  010410  062706  000010             ADD     #10,SP
 (4)  010414                    L10041:
 (4)  010414  104423                     TRAP    C$MSG
5589  010416                             $MD     36,2,2
 (4)  010416                    ERR36::
 (9)  010416  012746  005734             MOV     #DH2,-(SP)
 (8)  010422  012746  004100             MOV     #FM1,-(SP)
```

```
   (7)  010426  012746  000002                    MOV     #2,-(SP)
   (4)  010432  010600                            MOV     SP,R0
   (5)  010434  104414                            TRAP    C$PNTB
   (5)  010436  062706  000006                    ADD     #6,SP
   (11) 010442  013746  002640                    MOV     $BDDAT,-(SP)
   (10) 010446  013746  002636                    MOV     $GDDAT,-(SP)
   (9)  010452  012746  004132                    MOV     #TFM2,-(SP)
   (8)  010456  012746  000003                    MOV     #3,-(SP)
   (5)  010462  010600                            MOV     SP,R0
   (6)  010464  104414                            TRAP    C$PNTB
   (6)  010466  062706  000010                    ADD     #10,SP
   (4)  010472                              L10042:
   (4)  010472  104423                            TRAP    C$MSG
 5590
 5591  010474                                     BGNMSG  ERR37
   (3)  010474                              ERR37::
 5592  010474                                     PRINTF  #FM1,#EM1
   (8)  010474  012746  004245                    MOV     #EM1,-(SP)
   (7)  010500  012746  004100                    MOV     #FM1,-(SP)
   (6)  010504  012746  000002                    MOV     #2,-(SP)
   (3)  010510  010600                            MOV     SP,R0
   (4)  010512  104417                            TRAP    C$PNTF
   (4)  010514  062706  000006                    ADD     #6,SP
 5593  010520                                     PRINTF  #TFM37,$GDADR
   (8)  010520  013746  002632                    MOV     $GDADR,-(SP)
   (7)  010524  012746  004211                    MOV     #TFM37,-(SP)
   (6)  010530  012746  000002                    MOV     #2,-(SP)
   (3)  010534  010600                            MOV     SP,R0
   (4)  010536  104417                            TRAP    C$PNTF
   (4)  010540  062706  000006                    ADD     #6,SP
 5594  010544                                     ENDMSG
   (3)  010544                              L10043:
   (3)  010544  104423                            TRAP    C$MSG
 5595
 5596
 5597
```

```
 5599                                   .SBTTL  REPORT CODING SECTION
 5600
 5601
 5602                                   ;++
 5603                                   ; THE REPORT CODING SECTION CONTAINS THE
 5604                                   ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
 5605                                   ;--
 5606
 5607  010546                                   BGNRPT
  (3)  010546                           L$RPT::
 5608
 5614
 5615  010546                                   EXIT    RPT
  (4)  010546  000167                      ,WORD   J$JMP
  (3)  010550  000000                      .WORD   L10044-2-.
 5616
 5623
 5624  010552                                   ENDRPT
  (3)  010552                           L10044:
  (3)  010552  104425                      TRAP    C$RPT
 5625
 5626
 5627
 5628
 5629
```

```
                                .SBTTL  INITIALIZE SECTION

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

010554                                  BGNINIT
010554                          L$INIT::

                                ;INITIALIZE SUBROUTINE STACK
010554  012705  003130                  MOV     #SSTACK,R5
                                ;STORE BASE LEVEL PROGRAM STACK POINTER
010560  010637  002554                  MOV     SP,PSTACK
010564  005737  002646                  TST     FTIME
010570  001011                          BNE     1$
010572  013737  000004  002650          MOV     @#4,SAVE4
010600  013737  000006  002652          MOV     @#6,SAVE6
010606  012737  000001  002646          MOV     #1,FTIME
010614  013737  002650  000004  1$:     MOV     SAVE4,@#4
010622  013737  002652  000006          MOV     SAVE6,@#6
                                ;SEE IF PROGRAM JUST STARTED, BR IF YES
010630                                  READEF  #EF.START
010630  012700  000040                  MOV     #EF.START,R0
010634  104447                          TRAP    C$REFG
010636                                  BCOMPLETE       NEWST
010636  103414                          BCS     NEWST
                                ;SEE IF THIS IS A NEW PASS, BR IF YES
010640                                  READEF  #EF.NEW
010640  012700  000035                  MOV     #EF.NEW,R0
010644  104447                          TRAP    C$REFG
010646                                  BCOMPLETE       NEWST
010646  103410                          BCS     NEWST
                                ;SEE IF PROGRAM WAS JUST CONTINUED
010650                                  READEF  #EF.CONTINUE
010650  012700  000036                  MOV     #EF.CONTINUE,R0
010654  104447                          TRAP    C$REFG
010656                                  BCOMPLETE       ENDIT
010656  103570                          BCS     ENDIT

                                ;SEE IF PROGRAM JUST RESTARTED, BR IF NOT
010660                                  READEF  #EF.RESTART
010660  012700  000037                  MOV     #EF.RESTART,R0
010664  104447                          TRAP    C$REFG
010666                                  BNCOMPLETE      GETPRM
010666  103003                          BCC     GETPRM
010670                          NEWST:
                                ;RESET LOGICAL DEVICE TO -1
010670  012737  177777  002552          MOV     #-1,LOGDEV

                                ;GET UNIBUS ADRS, VECTOR, PRIORITY LEVEL, LINE UNIT,SWITCH
                                ;    PACKS, TEST CONNECTOR INFO. FOR THIS M8200,4,7 (CURRENT LOGICAL
                                ;    DEVICE).
010676                          GETPRM:
010676  005237  002552                  INC     LOGDEV
010702  023737  002552  002012          CMP     LOGDEV,L$UNIT
```

```
5674  010710  002367                        BGE     NEWST
5675  010712                                GPHARD - LOGDEV,R1
 (3)  010712  013700  002552                MOV     LOGDEV,R0
 (3)  010716  104442                        TRAP    C$GPHRD
 (3)  010720  010001                        MOV     R0,R1
5676  010722                                BNCOMPLETE      GETPRM
 (2)  010722  103365                        BCC     GETPRM
5677                                  ;GET ADDRESS OF M8200,4,7
5678  010724  012137  002626                MOV     (R1)+,WTYPE
5679  010730  011137  002716                MOV     (R1),KMCSR
5680                                  ;GET POINTER TO M8200,4,7 CSR HI BYTE
5681  010734  011137  002720                MOV     (R1),KMCSRH
5682  010740  005237  002720                INC     KMCSRH
5683                                  ;GET POINTER TO M8200,4,7 CTL OUT REG
5684  010744  011137  002722                MOV     (R1),KMCTL
5685  010750  062737  000002  002722        ADD     #2,KMCTL
5686                                  ;GET POINTER TO M8200,4,7 PORT REG - SEL 4
5687  010756  011137  002724                MOV     (R1),KMP04
5688  010762  062737  000004  002724        ADD     #4,KMP04
5689                                  ;GET POINTER TO M8200,4,7 PORT REG - SEL 6
5690  010770  012137  002726                MOV     (R1)+,KMP06
5691  010774  062737  000006  002726        ADD     #6,KMP06
5692                                  ;GET POINTER TO RCV VECTOR
5693  011002  011137  002706                MOV     (R1),KMRVEC
5694                                  ;GET POINTER TO RCV PRIORITY LEVEL
5695  011006  011137  002710                MOV     (R1),KMRLVL
5696  011012  062737  000002  002710        ADD     #2,KMRLVL
5697                                  ;GET POINTER TO TX VECTOR
5698  011020  011137  002712                MOV     (R1),KMTVEC
5699  011024  062737  000004  002712        ADD     #4,KMTVEC
5700                                  ;GET POINTER TO TX PRIORITY LEVEL
5701  011032  011137  002714                MOV     (R1),KMTLVL
5702  011036  062737  000006  002714        ADD     #6,KMTLVL
5703                                  ;PUT VECTOR INTO STAT1
5704  011044  012137  002700                MOV     (R1)+,STAT1
5705                                  ;PUT PRIORITY INTO STAT1
5706  011050  052137  002700                BIS     (R1)+,STAT1
5707                                  ;SEE IF NO LINE UNIT, SET BIT IF YES
5708  011054  005711                        TST     (R1)
5709  011056  001004                        BNE     50000$
5710  011060  052737  010000  002700        BIS     #BIT12,STAT1
5711  011066  000416                        BR      4$
5712  011070                                50000$:
5713                                  ;SEE IF M8201 LINE UNIT, SET BIT IF YES
5714  011070  021127  000001                CMP     (R1),#1
5715  011074  001001                        BNE     50001$
5716  011076  000412                        BR      4$
5717  011100                                50001$:
5718                                  ;SEE IF M8202 LINE UNIT, SET BIT IF YES
5719  011100  021127  000002                CMP     (R1),#2
5720  011104  001004                        BNE     50002$
5721  011106  052737  020000  002700        BIS     #BIT13,STAT1
5722  011114  000403                        BR      4$
5723  011116                                50002$:
5724                                  ;SET BIT FOR M8203 LINE UNIT
5725  011116  052737  100000  002700        BIS     #BIT15,STAT1
```

```
 5726  011124                                  4$:
 5727                                          ;SET BIT IN STAT1 FOR TEST CONNECTOR
 5728  011124  056137  000006  002700          BIS     6(R1),STAT1
 5729  011132  062701  000002                  ADD     #2,R1
 5730                                          ;SET SWITCH PACK #1 IN STAT2 LOW BYTE
 5731  011136  012137  002702                  MOV     (R1)+,STAT2
 5732                                          ;SET SWITCH PACK #2 IN STAT2 HIGH BYTE
 5733  011142  111137  002703                  MOVB    (R1),STAT2+1
 5734
 5735                                          ;INCREMENT LOGICAL UNIT (DEVICE) NUMBER
 5736                                          ;       INC     LOGDEV
 5737  011146  000240                          NOP
 5738  011150  000240                          NOP
 5739
 5740  011152  012737  002000  002606          MOV     #2000,MEMSZ
 5741  011160  005037  002630                  CLR     TYPE
 5742  011164  123727  002626  000000          CMPB    WTYPE,#0
 5743  011172  001422                          BEQ     ENDIT
 5744  011174  123727  002626  000004          CMPB    WTYPE,#4        ;KMC?
 5745  011202  001004                          BNE     5$
 5746  011204  012737  000001  002630          MOV     #1,TYPE
 5747  011212  000412                          BR      ENDIT
 5748  011214  012737  003777  002606  5$:     MOV     #3777,MEMSZ
 5749  011222  123727  002626  000006          CMPB    WTYPE,#6
 5750  011230  001003                          BNE     ENDIT
 5751  011232  012737  000001  002630          MOV     #1,TYPE
 5752  011240                          ENDIT:
 5753  011240                                  ENDINIT
  (3)  011240                          L10045:
  (3)  011240  104411                          TRAP    C$INIT
 5754
 5755                                  .EVEN
 5756  011242                                  BGNAUTO
  (3)  011242                          L$AUTO::
 5757                                          ;DEVICE DOES NOT HAVE A 'READY'
 5758  011242  013701  002716                  MOV     KMCSR,R1        ;R1 CONTAINS BASE M8200,4,7 ADDRESS
 5759  011246  012705  000004                  MOV     #4,R5           ;4 REGISTERS TO BE TESTED
 5760  011252  012737  011304  000004          MOV     #2$,4           ;SET UP TIMEOUT TRAP
 5761  011260  012737  000340  000006          MOV     #340,6          ;LEVEL 7
 5762  011266  005711                  1$:     TST     (R1)            ;REFERENCE DEVICE REGISTER
 5763  011270  000240                          NOP
 5764  011272  062701  000002                  ADD     #2,R1           ;NEXT REGISTER
 5765  011276  005305                          DEC     R5              ;DEC REGISTER COUNT
 5766  011300  001372                          BNE     1$              ;BR IF NOT LAST REGISTER
 5767  011302  000407                          BR      3$
 5768  011304  062706  000004          2$:     ADD     #4,SP
 5769  011310  010137  002632                  MOV     R1,$GDADR
 5770  011314                                  DODU    LOGDEV
  (3)  011314  013700  002552                  MOV     LOGDEV,R0
  (3)  011320  104451                          TRAP    C$DODU
 5771
 5772  011322  013737  002650  000004  3$:     MOV     SAVE4,4
 5773  011330  013737  002652  000006          MOV     SAVE6,6
 5774  011336                                  ENDAUTO
  (3)  011336                          L10046:
  (3)  011336  104461                          TRAP    C$AUTO
```

```
5775
```

```
 5777                                   .SBTTL  CLEANUP CODING SECTION
 5778
 5779                                   ;////////////////////////////////////////////////////////////////////////
 5780                                   ;/ THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
 5781                                   ;/ AT THE END OF EACH PASS.
 5782                                   ;////////////////////////////////////////////////////////////////////////
 5783
 5784  011340                                   BGNCLN
  (3)  011340                           L$CLEAN::
 5785  011340                                   BRESET
  (3)  011340  104433                      TRAP    C$RESET
 5786
 5787  011342                                   ENDCLN
  (3)  011342                           L10047:
  (3)  011342  104412                      TRAP    C$CLEAN
 5788
 5789
 5790
 5791
 5792
```

```
5794                                    .SBTTL  DROP UNIT SECTION
5795
5796                                    ;////////////////////////////////////////////////////////////////////////////////
5797                                    ;/ THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
5798                                    ;/ TO NO LONGER BE TESTED.
5799                                    ;////////////////////////////////////////////////////////////////////////////////
5800
5801  011344                                    BGNDU
 (3)  011344                            L$DU::
5802                                    ;ISSUE UNIBUS RESET TO CLEAN UP     .
5803  011344                                    BRESET
 (3)  011344  104433                        TRAP    C$RESET
5804  011346                                    ENDDU
 (3)  011346                            L10050:
 (3)  011346  104453                        TRAP    C$DU
5805
5806
5807
5808
5809
```

```
 5811                                   .SBTTL  ADD UNIT SECTION
 5812
 5813                                   ;////////////////////////////////////////////////////////////////////////
 5814                                   ;/ THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
 5815                                   ;/  TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING.  IF
 5816                                   ;/  'EF.AUNIT' IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
 5817                                   ;////////////////////////////////////////////////////////////////////////
 5818
 5819  011350                                   BGNAU
  (3)  011350                           L$AU::
 5820  011350                                   ENDAU
  (3)  011350                           L10051:
  (3)  011350  104452                          TRAP    C$AU
 5821
 5822
 5823
 5824
 5825
 5826
```

```
                                .SBTTL  HARDWARE TESTS
                                        BADHEAD
                                        ;********************* TEST 1 *********************
                                        ;*VERIFY THAT REFERENCING UNIBUS DEVICE REGISTERS
                                        ;*DOES NOT CAUSE A TIME OUT TRAP
                                        BADHEAD
                                        ;********************* TEST 1 *********************
                                BGNTST
                                T1::
011352  013701  002716                  MOV     KMCSR,R1        ;R1 CONTAINS BASE M8200,4,7 ADDRESS
011356  012705  000004                  MOV     #4,R5           ;4 REGISTERS TO BE TESTED
011362  012737  011420  000004          MOV     #2$,4           ;SET UP TIMEOUT TRAP
011370  012737  000340  000006          MOV     #340,6          ;LEVEL 7
011376  005711                  1$:     TST     (R1)            ;REFERENCE DEVICE REGISTER
011400  000240                          NOP
011402                                  ESCAPE TST
011402  104410                          TRAP    C$ESCAPE
011404  000072                          .WORD   L10052-.
011406  062701  000002                  ADD     #2,R1           ;NEXT REGISTER
011412  005305                          DEC     R5              ;DEC REGISTER COUNT
011414  001370                          BNE     1$              ;BR IF NOT LAST REGISTER
011416  000417                          BR      3$
011420  062706  000004          2$:     ADD     #4,SP
011424  010137  002632                  MOV     R1,$GDADR
011430                                  ERROR   37              ;TIME-OUT ERROR
011446  104455                          TRAP    C$ERDF
011450  000045                          .WORD   37
011452  005701                          .WORD   EM37
011454  010474                          .WORD   ERR37

011456  013737  002650  000004  3$:     MOV     SAVE4,4
011464  013737  002652  000006          MOV     SAVE6,6
011472                                  ESCAPE  TST
011472  104410                          TRAP    C$ESCAPE
011474  000002                          .WORD   L10052-.
011476                          ENDTST
011476                          L10052:
011476  104401                          TRAP    C$ETST

011500                                  BADHEAD
                                        ;********************* TEST 2 *********************
                                        ;*VERIFY THAT RUN CAN BE CLEARED
011500                                  BADHEAD
                                        ;********************* TEST 2 *********************
011500                          BGNTST
011500                          T2::
011500                                  MVINT
011500  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
011504  005011                          CLR     (R1)            ;CLEAR KMCSR
011506  005002                          CLR     R2              ;CLEAR 'EXPECTED''
```

```
 5867  011510  011104                        MOV     (R1),R4         ;PUT KMCSR IN ''FOUND''
 5868  011512  001413                        BEQ     1$              ;BR IF CLEARED
 5869  011514                                ERROR   26              ;ERROR KMCSR NOT CLEARED
  (5)  011532  104455                        TRAP    C$ERDF
  (6)  011534  000032                        .WORD   26
  (6)  011536  005460                        .WORD   EM26
  (6)  011540  007552                        .WORD   ERR26
 5870  011542                          1$:
 5871  011542                          ENDTST
  (3)  011542                          L10053:
  (3)  011542  104401                        TRAP    C$ETST
 5872
 5873  011544                                BADHEAD
  (2)                                       ;******************** TEST 3 ********************
 5874                                       ;*UNIBUS REGISTER WORD DUAL ADDRESSING TEST
 5875                                       ;*LOAD ALL REGISTERS WITH INCREMENTING PATTERN
 5876                                       ;*READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING
 5877                                       ;*THE SEQUENCE:
 5878                                       ;*      1. CLEAR REGISTER
 5879                                       ;*      2. WRITE PATTERN
 5880                                       ;*      3. VERIFY PATTERN
 5881                                       ;*      4. DO ALL 4 REGISTERS
 5882                                       ;*      5. READ ALL BACK IF ERRORS,
 5883                                       ;*         DUAL ADDRESS PROBLEM.
 5884                                       ;*
 5885                                       ;*         1 IN REG 0
 5886                                       ;*         2 IN REG 2
 5887                                       ;*         3 IN REG 4
 5888                                       ;*         4 IN REG 6
 5889  011544                                BADHEAD
  (2)                                       ;******************** TEST 3 ********************
 5890
 5891  011544                          BGNTST
  (3)  011544                          T3::
 5892  011544                                MYINT
  (1)  011544  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 5893                                                                  ;R1 CONTAINS BASE M8200,4,7 ADDRESS
 5894  011550                                MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  011550  004537  003142                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 5895  011554  012702  000001                MOV     #1,R2           ;START PATTERN AT 1
 5896  011560                                BGNSEG
  (3)  011560  104404                        TRAP    C$BSEG
 5897  011562  005011                  1$:   CLR     (R1)            ;CLEAR REGISTER
 5898  011564  010211                        MOV     R2,(R1)         ;WRITE M8200,4,7 REGISTER WITH PATTERN
 5899  011566  011104                        MOV     (R1),R4         ;READ M8200,4,7 REGISTER INTO ''FOUND''
 5900  011570  020204                        CMP     R2,R4           ;IS DATA CORRECT
 5901  011572  001413                        BEQ     2$              ;BR IS YES
 5902  011574                                ERROR   26              ;DATA ERROR
  (5)  011612  104455                        TRAP    C$ERDF
  (6)  011614  000032                        .WORD   26
  (6)  011616  005460                        .WORD   EM26
  (6)  011620  007552                        .WORD   ERR26
 5903  011622                          2$:   ESCAPE  SEG
  (3)  011622  104410                        TRAP    C$ESCAPE
  (3)  011624  000014                        .WORD   10000$-.
 5904  011626  005721                        TST     (R1)+           ;NEXT REGISTER
```

```
 5905  011630  005202
 5906  011632  022702  000005
 5907  011636  001351
 5908  011640
  (3)  011640
  (3)  011640  104405
 5909  011642  013701  002716
 5910  011646  012702  000001
 5911  011652
  (3)  011652  104404
 5912  011654
 5913  011654  011104
 5914  011656  020204
 5915  011660  001413
 5916  011662
  (5)  011700  104455
  (6)  011702  000002
  (6)  011704  004273
  (6)  011706  006116
 5917  011710
  (3)  011710  104410
  (3)  011712  000014
 5918  011714  005721
 5919  011716  005202
 5920  011720  022702  000005
 5921  011724  001353
 5922  011726
  (3)  011726
  (3)  011726  104405
 5923  011730
  (3)  011730
  (3)  011730  104401
 5924
 5925  011732
  (2)
 5926
 5927
 5928
 5929  011732
  (2)
 5930
 5931  011732
  (3)  011732
 5932  011732
  (1)  011732  004537  003142
 5933  011736  005037  002624
 5934  011742  012702  000001
 5935  011746
  (3)  011746  104404
 5936  011750  013701  002716
 5937  011754  010237  002636
 5938  011760  013711  002636
 5939  011764  011104
 5940  011766  023704  002636
 5941  011772  001411
 5942  011774
```

```
                INC     R2              ;INCREMENT DATA PATTERN
                CMP     #5,R2           ;LAST REGISTER?
                BNE     1$              ;BR IF NO
                ENDSEG
10000$:
                TRAP    C$ESEG
                MOV     KMCSR,R1        ;BASE M8200,4,7 ADDRESS TO R1
                MOV     #1,R2           ;RESTART PATTERN AT 1
                BGNSEG
                TRAP    C$BSEG
3$:
                MOV     (R1),R4         ;READ COMM. MICR-PROCESSOR FAMILY REGISTER INTO 'FOUND'
                CMP     R2,R4           ;IS DATA CORRECT
                BEQ     4$              ;BR IF YES
                ERROR   2               ;DUAL ADDRESSING ERROR
                TRAP    C$ERDF
                .WORD   2
                .WORD   EM2
                .WORD   ERR2
4$:             ESCAPE  SEG
                TRAP    C$ESCAPE
                .WORD   10001$-.
                TST     (R1)+           ;NEXT REGISTER
                INC     R2              ;INCREMENT PATTERN
                CMP     #5,R2           ;LAST REGISTER?
                BNE     3$              ;BR IF NO
                ENDSEG
10001$:
                TRAP    C$ESEG
ENDTST
L10054:
                TRAP    C$ETST

                BADHEAD
                ;******************** TEST 4 ********************
                ;*CONTROL STATUS REGISTER WRITE/READ TEST
                ;*FLOAT A ONE THROUGH BSEL 0
                ;*CLEAR BIT0, VERIFY BIT0 WAS CLEARED
                BADHEAD
                ;******************** TEST 4 ********************

BGNTST
T4::
                MSTCLR                          ;MASTER CLEAR M8200,4,7
                JSR     R5,.MSTCLR                      ;CLEAR M8200,4,7
                CLR     MR0
                MOV     #BIT0,R2        ;INDICATE BSEL0
                BGNSEG
                TRAP    C$BSEG
1$:             MOV     KMCSR,R1        ;PUT REGISTER ADDRESS IN R1
                MOV     R2,$GDDAT
                MOV     $GDDAT,(R1)                     ;WRITE BIT 0
                MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
                CMP     $GDDAT,R4               ;IS DATA CORRECT
                BEQ     2$              ;BR IF YES
                ERROR   27,YES          ;DATA ERROR
```

```
  (5)  012006  104455                    TRAP    C$ERDF
  (6)  012010  000033                    .WORD   27
  (6)  012012  005511                    .WORD   EM27
  (6)  012014  007630                    .WORD   ERR27
 5943  012016                    2$:     ESCAPE  SEG
  (3)  012016  104410                    TRAP    C$ESCAPE
  (3)  012020  000052                    .WORD   10000$-.
 5944  012022  040211                    BIC     R2,(R1)         ;CLEAR BSEL0
 5945  012024  005037  002636            CLR     $GDDAT          ;CLEAR 'EXPECTED'
 5946  012030  011104                    MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
 5947  012032  001413                    BEQ     4$              ;BR IF ZERO
 5948  012034                            ERROR   2               ;DATA ERROR BSEL NOT CLEARED
  (5)  012052  104455                    TRAP    C$ERDF
  (6)  012054  000002                    .WORD   2
  (6)  012056  004273                    .WORD   EM2
  (6)  012060  006116                    .WORD   ERR2
 5949  012062                    4$:     ESCAPE  SEG
  (3)  012062  104410                    TRAP    C$ESCAPE
  (3)  012064  000006                    .WORD   10000$-.
 5950  012066  106302                    ASLB    R2
 5951  012070  001327                    BNE     1$
 5952  012072                            ENDSEG
  (3)  012072                    10000$:
  (3)  012072  104405                    TRAP    C$ESEG
 5953  012074                    ENDTST
  (3)  012074                    L10055:
  (3)  012074  104401                    TRAP    C$ETST
 5954
 5955
```

```
 5957
 5958
 5959
 5960
 5961
 5962  012076                                  BADHEAD
  (2)                                          ;******************** TEST 5 ********************
 5963                                          ;*CONTROL STATUS REGISTER WRITE/READ TEST
 5964                                          ;*SET BIT9, VERIFY BIT9 WAS SET
 5965                                          ;*CLEAR BIT9, VERIFY BIT9 WAS CLEARED
 5966  012076                                  BADHEAD
  (2)                                          ;******************** TEST 5 ********************
 5967
 5968  012076                          BGNTST
  (3)  012076                          T5::
 5969  012076                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  012076  004537  003142                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 5970  012102                                  BGNSEG
  (3)  012102  104404                        TRAP    C$BSEG
 5971  012104  013701  002716          1$:   MOV     KMCSR,R1          ;PUT REGISTER ADDRESS IN R1
 5972  012110  012702  001000                MOV     #BIT9,R2          ;PUT DATA IN 'EXPECTED''
 5973  012114  010211                        MOV     R2,(R1)           ;WRITE BIT 9
 5974  012116  011104                        MOV     (R1),R4           ;READ CONTROL STATUS REGISTER
 5975  012120  020204                        CMP     R2,R4             ;IS DATA CORRECT
 5976  012122  001413                        BEQ     2$                ;BR IF YES
 5977  012124                                ERROR   26                ;DATA ERROR
  (5)  012142  104455                        TRAP    C$ERDF
  (6)  012144  000032                        .WORD   26
  (6)  012146  005460                        .WORD   EM26
  (6)  012150  007552                        .WORD   ERR26
 5978  012152                          2$:   ESCAPE  SEG
  (3)  012152  104410                        TRAP    C$ESCAPE
  (3)  012154  000002                        .WORD   10000$-.
 5979  012156                                ENDSEG
  (3)  012156                          10000$:
  (3)  012156  104405                        TRAP    C$ESEG
 5980  012160                                BGNSEG
  (3)  012160  104404                        TRAP    C$BSEG
 5981  012162  042711  001000          3$:   BIC     #BIT9,(R1)        ;CLEAR BIT 9
 5982  012166  005002                        CLR     R2                ;CLEAR 'EXPECTED''
 5983  012170  011104                        MOV     (R1),R4           ;READ CONTROL STATUS REGISTER
 5984  012172  001416                        BEQ     4$                ;BR IF ZERO
 5985  012174                                ERROR   26                ;DATA ERROR BIT9 NOT CLEARED
  (5)  012212  104455                        TRAP    C$ERDF
  (6)  012214  000032                        .WORD   26
  (6)  012216  005460                        .WORD   EM26
  (6)  012220  007552                        .WORD   ERR26
 5986  012222                                ESCAPE  SEG
  (3)  012222  104410                        TRAP    C$ESCAPE
  (3)  012224  000002                        .WORD   10001$-.
 5987  012226                                ENDSEG
  (3)  012226                          10001$:
  (3)  012226  104405                        TRAP    C$ESEG
 5988  012230                          4$:
 5989  012230                          ENDTST
  (3)  012230                          L10056:
```

```
  (3)  012230  104401                        TRAP    C$ETST
 5990
 5991  012232                                BADHEAD
  (2)                                        ;******************** TEST 6 ********************
 5992                                        ;*CONTROL STATUS REGISTER WRITE/READ TEST
 5993                                        ;*SET BIT11, VERIFY BIT11 WAS SET
 5994                                        ;*CLEAR BIT11, VERIFY BIT11 WAS CLEARED
 5995  012232                                BADHEAD
  (2)                                        ;******************** TEST 6 ********************
 5996
 5997  012232                          BGNTST
  (3)  012232                          T6::
 5998  012232                                MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  012232  004537  003142                 JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 5999  012236                                BGNSEG
  (3)  012236  104404                        TRAP    C$BSEG
 6000  012240  013701  002716          1$:   MOV     KMCSR,R1          ;PUT REGISTER ADDRESS IN R1
 6001  012244  012702  004000                MOV     #BIT11,R2         ;PUT DATA IN 'EXPECTED'
 6002  012250  010211                        MOV     R2,(R1)           ;WRITE BIT 11
 6003  012252  011104                        MOV     (R1),R4           ;READ CONTRCL STATUS REGISTER
 6004  012254  020204                        CMP     R2,R4             ;IS DATA CORRECT
 6005  012256  001413                        BEQ     2$                ;BR IF YES
 6006  012260                                ERROR   26                ;DATA ERROR
  (5)  012276  104455                        TRAP    C$ERDF
  (6)  012300  000032                        .WORD   26
  (6)  012302  005460                        .WORD   EM26
  (6)  012304  007552                        .WORD   ERR26
 6007  012306                          2$:   ESCAPE  SEG
  (3)  012306  104410                        TRAP    C$ESCAPE
  (3)  012310  000002                        .WORD   10000$-.
 6008  012312                                ENDSEG
  (3)  012312                          10000$:
  (3)  012312  104405                        TRAP    C$ESEG
 6009  012314                                BGNSEG
  (3)  012314  104404                        TRAP    C$BSEG
 6010  012316  042711  004000          3$:   BIC     #BIT11,(R1)       ;CLEAR BIT 11
 6011  012322  005002                        CLR     R2                ;CLEAR 'EXPECTED'
 6012  012324  011104                        MOV     (R1),R4           ;READ CONTROL STATUS REGISTER
 6013  012326  001414                        BEQ     4$                ;BR IF ZERO
 6014  012330                                ERROR   26                ;DATA ERROR BIT11 NOT CLEARED
  (5)  012346  104455                        TRAP    C$ERDF
  (6)  012350  000032                        .WORD   26
  (6)  012352  005460                        .WORD   EM26
  (6)  012354  007552                        .WORD   ERR26
 6015  012356                                ENDSEG
  (3)  012356                          10001$:
  (3)  012356  104405                        TRAP    C$ESEG
 6016  012360                          4$:
 6017  012360                          ENDTST
  (3)  012360                          L10057:
  (3)  012360  104401                        TRAP    C$FTST
 6018
 6019  012362                                BADHEAD
  (2)                                        ;******************** TEST 7 ********************
 6020                                        ;*CONTROL STATUS REGISTER WRITE/READ TEST
 6021                                        ;*SET BIT12, VERIFY BIT12 WAS SET
```

```
                                        ;*CLEAR BIT12, VERIFY BIT12 WAS CLEARED
                                        BADHEAD
                                        ;******************** TEST 7 ********************

                                BGNTST
                                T7::
                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
                                (1)     JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
                                        BGNSEG
                                (3)     TRAP    C$BSEG
                                1$:     MOV     KMCSR,R1        ;PUT REGISTER ADDRESS IN R1
                                        MOV     #BIT12,R2       ;PUT DATA IN 'EXPECTED'
                                        MOV     R2,(R1)         ;WRITE BIT 12
                                        MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
                                        CMP     R2,R4           ;IS DATA CORRECT
                                        BEQ     2$              ;BR IF YES
                                        ERROR   26              ;DATA ERROR
                                (5)     TRAP    C$ERDF
                                (6)     .WORD   26
                                (6)     .WORD   EM26
                                (6)     .WORD   ERR26
                                2$:     ESCAPE  SEG
                                (3)     TRAP    C$ESCAPE
                                (3)     .WORD   10000$-.
                                        ENDSEG
                                10000$:
                                (3)     TRAP    C$ESEG
                                        BGNSEG
                                (3)     TRAP    C$BSEG
                                3$:     BIC     #BIT12,(R1)     ;CLEAR BIT 12
                                        CLR     R2              ;CLEAR 'EXPECTED'
                                        MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
                                        BEQ     4$              ;BR IF ZERO
                                        ERROR   26              ;DATA ERROR BIT12 NOT CLEARED
                                (5)     TRAP    C$ERDF
                                (6)     .WORD   26
                                (6)     .WORD   EM26
                                (6)     .WORD   ERR26
                                        ENDSEG
                                10001$:
                                (3)     TRAP    C$ESEG
                                4$:
                                ENDTST
                                L10060:
                                (3)     TRAP    C$ETST

                                        BADHEAD
                                (2)     ;******************** TEST 8 ********************
                                        ;*CONTROL OUT REGISTER WRITE/READ TEST
                                        ;*FLOAT A ONE THROUGH SEL2
                                        BADHEAD
                                (2)     ;******************** TEST 8 ********************

                                BGNTST
                                T8::
                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
```

```
  (1)  012512  004537  003142                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 6054  012516  012737  000002  002624        MOV     #2,MR0
 6055  012524  012702  000001                MOV     #1,R2
 6056  012530                                BGNSEG
  (3)  012530  104404                        TRAP    C$BSEG
 6057
 6058  012532  013701  002722          1$:   MOV     KMCTL,R1          ;PUT REGISTER ADDRESS IN R1
 6059  012536  010237  002636                MOV     R2,$GDDAT         ;PUT DATA IN 'EXPECTED'
 6060  012542  013711  002636                MOV     $GDDAT,(R1)               ;WRITE BIT 0
 6061  012546  011104                        MOV     (R1),R4           ;READ CONTROL OUT REGISTER
 6062  012550  023704  002636                CMP     $GDDAT,R4                 ;IS DATA CORRECT
 6063  012554  001411                        BEQ     2$                ;BR IF YES
 6064  012556                                ERROR   27,YES            ;DATA ERROR
  (5)  012570  104455                        TRAP    C$ERDF
  (6)  012572  000033                        .WORD   27
  (6)  012574  005511                        .WORD   EM27
  (6)  012576  007630                        .WORD   ERR27
 6065  012600                          2$:   ESCAPE  SEG
  (3)  012600  104410                        TRAP    C$ESCAPE
  (3)  012602  000046                        .WORD   10000$-.
 6066  012604  040211                  3$:   BIC     R2,(R1) ;CLEAR BIT
 6067  012606  005037  002636                CLR     $GDDAT            ;CLEAR 'EXPECTED'
 6068  012612  011104                        MOV     (R1),R4           ;READ CONTROL OUT REGISTER
 6069  012614  001411                        BEQ     4$                ;BR IF ZERO
 6070  012616                                ERROR   27,YES            ;DATA ERROR BIT0 NOT CLEARED
  (5)  012630  104455                        TRAP    C$ERDF
  (6)  012632  000033                        .WORD   27
  (6)  012634  005511                        .WORD   EM27
  (6)  012636  007630                        .WORD   ERR27
 6071  012640                          4$:   ESCAPEE SEG
  (3)  012640  104410                        TRAP    C$ESCAPE
  (3)  012642  000006                        .WORD   10000$-.
 6072  012644  006302                        ASL     R2
 6073  012646  001331                        BNE     1$
 6074  012650                                ENDSEG
  (3)  012650                          10000$:
  (3)  012650  104405                        TRAP    C$ESEG
 6075  012652                          ENDTST
  (3)  012652                          L10061:
  (3)  012652  104401                        TRAP    C$ETST
 6076
 6077
 6078
 6079
 6080
 6081
 6082  012654                                BADHEAD
  (2)                                        ;********************* TEST 9 *********************
 6083                                        ;*PORT4 REGISTER WRITE/READ TEST
 6084                                        ;*FLOAT A ONE THROUGH PORT4 REGISTER
 6085                                        ;*FLOAT A ZERO THROUGH PORT4 REGISTER
 6086  012654                                BADHEAD
  (2)                                        ;********************* TEST 9 *********************
 6087
 6088
 6089  012654                          BGNTST
```

```
  (3)  012654                          T9::
6090   012654  012737  000004  002624          MOV     #4,MR0
6091   012662                          MSTCLR
  (1)  012662  004537  003142                  JSR     R5,.MSTCLR      ;MASTER CLEAR M8200,4,7
6092   012666  013701  002724                  MOV     KMP04,R1        ;CLEAR M8200,4,7
6093   012672  012702  000001                  MOV     #1,R2           ;PUT REGISTER ADDRESS IN R1
6094   012676                                  BGNSEG                  ;START WITH BIT0
  (3)  012676  104404                          TRAP    C$BSEG
6095   012700                          64$:
6096   012700  010211                          MOV     R2,(R1)         ;WRITE PORT4 REGISTER
6097   012702  011104                          MOV     (R1),R4         ;READ PORT4 REGISTER
6098   012704  020204                          CMP     R2,R4           ;COMPARE EXPECTED AND FOUND
6099   012706  001413                          BEQ     65$             ;BR IF OK
6100   012710                                  ERROR   27              ;WRITE/READ ERROR
  (5)  012726  104455                          TRAP    C$ERDF
  (6)  012730  000033                          .WORD   27
  (6)  012732  005511                          .WORD   EM27
  (6)  012734  007630                          .WORD   ERR27
6101   012736                          65$:    ESCAPE  SEG
  (3)  012736  104410                          TRAP    C$ESCAPE
  (3)  012740  000010                          .WORD   10000$-.
6102   012742  000241                          CLC                     ;CLEAR CARRY
6103   012744  006102                          ROL     R2              ;SHIFT TO NEXT BIT
6104   012746  001354                          BNE     64$             ;BR IF NOT DONE YET?
6105   012750                                  ENDSEG
  (3)  012750                          10000$:
  (3)  012750  104405                          TRAP    C$ESEG
6106   012752  012702  000001                  MOV     #1,R2           ;START WITH BIT0
6107   012756                                  BGNSEG
  (3)  012756  104404                          TRAP    C$BSEG
6108   012760                          66$:
6109   012760  005102                          COM     R2              ;CHANGE TO A FLOATING ZERO
6110   012762  010211                          MOV     R2,(R1)         ;WRITE PORT4 REGISTER
6111   012764  011104                          MOV     (R1),R4
6112   012766  020204                          CMP     R2,R4           ;COMPARE EXPECTED AND FOUND
6113   012770  001413                          BEQ     67$             ;BR IF OK
6114   012772                                  ERROR   27              ;WRITE/READ ERROR
  (5)  013010  104455                          TRAP    C$ERDF
  (6)  013012  000033                          .WORD   27
  (6)  013014  005511                          .WORD   EM27
  (6)  013016  007630                          .WORD   ERR27
6115   013020                          67$:    ESCAPE  SEG
  (3)  013020  104410                          TRAP    C$ESCAPE
  (3)  013022  000012                          .WORD   10001$-.
6116   013024  005102                          COM     R2              ;CHANGE BACK TO A FLOATING ONE
6117   013026  000241                          CLC                     ;CLEAR CARRY
6118   013030  006102                          ROL     R2              ;SHIFT TO NEXT BIT
6119   013032  001352                          BNE     66$             ;BR IF NOT DONE YET?
6120   013034                                  ENDSEG
  (3)  013034                          10001$:
  (3)  013034  104405                          TRAP    C$ESEG
6121   013036                          ENDTST
  (3)  013036                          L10062:
  (3)  013036  104401                          TRAP    C$ETST
6122
6123   013040                                  BADHEAD
```

```
 (2)                                        ;******************** TEST 10 ********************
6124                                        ;*PORT6 REGISTER WRITE/READ TEST
6125                                        ;*FLOAT A ONE THROUGH PORT6 REGISTER
6126                                        ;*FLOAT A ZERO THROUGH PORT6 REGISTER
6127  013040                                BADHEAD
 (2)                                        ;******************** TEST 10 ********************
6128
6129  013040                          BGNTST
 (3)  013040                          T10::
6130  013040  012737  000006  002624          MOV     #6,MR0
6131  013046                                   MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  013046  004537  003142                   JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
6132  013052  013701  002726                   MOV     KMP06,R1        ;PUT REGISTER ADDRESS IN R1
6133  013056  012702  000001                   MOV     #1,R2            ;START WITH BIT0
6134  013062                                   BGNSEG
 (3)  013062  104404                           TRAP    C$BSEG
6135  013064                          64$:
6136  013064  010211                          MOV     R2,(R1)         ;WRITE PORT6 REGISTER
6137  013066  011104                          MOV     (R1),R4         ;READ PORT6 REGISTER
6138  013070  020204                          CMP     R2,R4           ;COMPARE EXPECTED AND FOUND
6139  013072  001413                          BEQ     65$             ;BR IF OK
6140  013074                                   ERROR   27              ;WRITE/READ ERROR
 (5)  013112  104455                          TRAP    C$ERDF
 (6)  013114  000033                          .WORD   27
 (6)  013116  005511                          .WORD   EM27
 (6)  013120  007630                          .WORD   ERR27
6141  013122                          65$:    ESCAPE  SEG
 (3)  013122  104410                          TRAP    C$ESCAPE
 (3)  013124  000010                          .WORD   10000$-.
6142  013126  000241                          CLC                     ;CLEAR CARRY
6143  013130  006105                          ROL     R5              ;SHIFT TO NEXT BIT
6144  013132  001354                          BNE     64$             ;BR IF NOT DONE YET?
6145  013134                                   ENDSEG
 (3)  013134                          10000$:
 (3)  013134  104405                          TRAP    C$ESEG
6146  013136  012702  000001                  MOV     #1,R2           ;START WITH BIT0
6147  013142                                   BGNSEG
 (3)  013142  104404                          TRAP    C$BSEG
6148  013144                          66$:
6149  013144  005102                          COM     R2              ;CHANGE TO A FLOATING ZERO
6150
6151  013146  010211                          MOV     R2,(R1)         ;WRITE PORT6 REGISTER
6152  013150  011104                          MOV     (R1),R4         ;READ PORT6 REGISTER
6153  013152  020204                          CMP     R2,R4           ;COMPARE EXPECTED AND FOUND
6154  013154  001413                          BEQ     67$             ;BR IF OK
6155  013156                                   ERROR   27              ;WRITE/READ ERROR
 (5)  013174  104455                          TRAP    C$ERDF
 (6)  013176  000033                          .WORD   27
 (6)  013200  005511                          .WORD   EM27
 (6)  013202  007630                          .WORD   ERR27
6156  013204                          67$:    ESCAPE  SEG
 (3)  013204  104410                          TRAP    C$ESCAPE
 (3)  013206  000012                          .WORD   10001$-.
6157  013210  005102                          COM     R2              ;CHANGE BACK TO A FLOATING ONE
6158  013212  000241                          CLC                     ;CLEAR CARRY
6159  013214  006102                          ROL     R2              ;SHIFT TO NEXT BIT
```

```
 6160  013216  001352                        BNE     66$             ;BR IF NOT DONE YET?
 6161  013220                                ENDSEG
   (3) 013220                          10001$:
   (3) 013220  104405                        TRAP    C$ESEG
 6162  013222                          .ENDTST
   (3) 013222                          L10063:
   (3) 013222  104401                        TRAP    C$ETST
 6163
 6164  013224                                BADHEAD
   (2)                                     ;****************** TEST 11 ********************
 6165                                      ;*UNIBUS REGISTER BYTE DUAL ADDRESSING TEST
 6166                                      ;*LOAD ALL REGISTERS WITH INCREMENTING PATTERN
 6167                                      ;*READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING
 6168  013224                              BADHEAD
   (2)                                     ;****************** TEST 11 ********************
 6169
 6170  013224                          BGNTST
   (3) 013224                          T11::
 6171  013224                                MYINT
   (1) 013224  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6172  013230                                MSTCLR                  ;MASTER CLEAR M8200,4,7
   (1) 013230  004537  003142          :     JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 6173  013234  012702  000001                MOV     #1,R2           ;START PATTERN AT 1
 6174  013240                                BGNSEG
   (3) 013240  104404                        TRAP    C$BSEG
 6175  013242  105011                  1$:   CLRB    (R1)            ;CLEAR REGISTER
 6176  013244  110211                        MOVB    R2,(R1)         ;WRITE M8200,4,7 REGISTER WITH PATTERN
 6177  013246  111104                        MOVB    (R1),R4         ;READ M8200,4,7 REGISTER INTO 'FOUND'
 6178  013250  120204                        CMPB    R2,R4           ;IS DATA CORRECT
 6179  013252  001413                        BEQ     2$              ;BR IF YES
 6180  013254                                ERROR   2               ;DATA ERROR
   (5) 013272  104455                        TRAP    C$ERDF
   (6) 013274  000002                        .WORD   2
   (6) 013276  004273                        .WORD   EM2
   (6) 013300  006116                        .WORD   ERR2
 6181  013302                          2$:   ESCAPE  SEG
   (3) 013302  104410                        TRAP    C$ESCAPE
   (3) 013304  000024                        .WORD   10000$-.
 6182  013306  105721                        TSTB    (R1)+           ;NEXT REGISTER
 6183  013310  005202                        INC     R2              ;INCREMENT DATA PATTERN
 6184  013312  022702  000011                CMP     #11,R2          ;LAST REGISTER?
 6185  013316  001351                        BNE     1$              ;BR IF NO
 6186  013320  013701  002716                MOV     KMCSR,R1        ;BASE M8200,4,7 ADDRESS TO R1
 6187  013324  012702  000001                MOV     #1,R2           ;RESTART PATTERN AT 1
 6188  013330                                ENDSEG
   (3) 013330                          10000$:
   (3) 013330  104405                        TRAP    C$ESEG
 6189  013332                                BGNSEG
   (3) 013332  104404                        TRAP    C$BSEG
 6190  013334                          3$:
 6191  013334  111104                        MOVB    (R1),R4         ;READ COMM.MICRO-PROCESSOR FAMILY REGISTER INTO 'FOUND'
 6192  013336  120204                        CMPB    R2,R4           ;IS DATA CORRECT
 6193  013340  001413                        BEQ     4$              ;BR IF YES
 6194  013342                                ERROR   2               ;DUAL ADDRESSING ERROR
   (5) 013360  104455                        TRAP    C$ERDF
   (6) 013362  000002                        .WORD   2
```

```
   (6)  013364  004273                        .WORD   EM2
   (6)  013366  006116                        .WORD   ERR2
 6195   013370                        4$:     ESCAPE  SEG
   (3)  013370  104410                        TRAP    C$ESCAPE
   (3)  013372  000014                        .WORD   10001$-.
 6196   013374  105721                        TSTB    (R1)+           ;NEXT REGISTER
 6197   013376  005202                        INC     R2              ;INCREMENT PATTERN
 6198   013400  022702  000011                CMP     #11,R2          ;LAST REGISTER?
 6199   013404  001353                        BNE     3$              ;BR IF NO
 6200   013406                                ENDSEG
   (3)  013406                        10001$:
   (3)  013406  104405                        TRAP    C$ESEG
 6201   013410                        ENDTST
   (3)  013410                        L10064:
   (3)  013410  104401                        TRAP    C$ETST
 6202
 6203   013412                                BADHEAD
   (2)                                         ;****************** TEST 12 ******************
 6204                                          ;*MAINTENANCE INSTRUCTION REGISTER TEST
 6205                                          ;*VERIFY THAT THE MAINT IR CAN BE WRITTEN TO ALL ZEROS'
 6206                                          ;*AND ALL ONES'. VERIFY THAT IT IS CLEARED ON A BUS RESET.
 6207   013412                                BADHEAD
   (2)                                         ;****************** TEST 12 ******************
 6208
 6209   013412                        BGNTST
   (3)  013412                        T12::
 6210
 6211   013412                                                        ;R1 CONTAINS BASE M8200,4,7 ADDRESS
                                               MSTCLR                ;MASTER CLEAR M8200,4,7
   (1)  013412  004537  003142                 JSR     R5,..MSTCLR           ;CLEAR M8200,4,7
 6212   013416                                MYINT
   (1)  013416  013701  002716                 MOV     KMCSR,R1      ;GET DEVICE ADDRESS.
 6213   013422                                BGNSEG
   (3)  013422  104404                        TRAP    C$BSEG
 6214   013424  012711  003000                MOV     #BIT9!BIT10,(R1) ;SEL6 IS NOW THE IR
 6215   013430  005002                        CLR     R2              ;PUT 'EXPECTED'' IN $GDDAT
 6216   013432  010261  000006        1$:     MOV     R2,6(R1)        ;CLEAR THE IR
 6217   013436  016104  000006                MOV     6(R1),R4        ;READ THE IR
 6218   013442  020204                        CMP     R2,R4           ;IS IT CLEARED?
 6219   013444  001413                        BEQ     2$              ;BR IF YES
 6220   013446                                ERROR   26              ;ERROR IR IS NOT CLEAR
   (5)  013464  104455                        TRAP    C$ERDF
   (6)  013466  000032                        .WORD   26
   (6)  013470  005460                        .WORD   EM26
   (6)  013472  007552                        .WORD   ERR26
 6221   013474                        2$:     ESCAPE  SEG
   (3)  013474  104410                        TRAP    C$ESCAPE
   (3)  013476  000002                        .WORD   10000$-.
 6222   013500                                ENDSEG
   (3)  013500                        10000$:
   (3)  013500  104405                        TRAP    C$ESEG
 6223   013502  012702  177777                MOV     #-1,R2          ;PUT 'EXPECTED'' IN $GDDAT
 6224   013506                                BGNSEG
   (3)  013506  104404                        TRAP    C$BSEG
 6225   013510  010261  000006        3$:     MOV     R2,6(R1)        ;WRITE ALL ONES TO THE IR
 6226   013514  016104  000006                MOV     6(R1),R4        ;READ THE IR
 6227   013520  020204                        CMP     R2,R4           ;IS IT ALL ONES?
```

```
 6228  013522  001413                    BEQ     4$              ;BR IF YES
 6229  013524                            ERROR   26              ;ERROR IR IS NOT - ALL ONES
  (5)  013542  104455                    TRAP    C$ERDF
  (6)  013544  000032                    .WORD   26
  (6)  013546  005460                    .WORD   EM26
  (6)  013550  007552                    .WORD   ERR26
 6230  013552                    4$:     ESCAPE  SEG
  (3)  013552  104410                    TRAP    C$ESCAPE
  (3)  013554  000002                    .WORD   10001$-.
 6231  013556                            ENDSEG
  (3)  013556                    10001$:
  (3)  013556  104405                    TRAP    C$ESEG
 6232  013560                    ENDTST
  (3)  013560                    L10065:
  (3)  013560  104401                    TRAP    C$ETST
 6233
 6234
```

```
                                                BADHEAD
                                                ;******************** TEST 13 ********************
                                                ;*MICRO PROCESSOR TEST
                                                ;*LOAD KMP06 WITH A MICRO-PROCESSOR INSTRUCTION, CLOCK IT
                                                ;*VERIFY INSTRUCTION EXECUTED PROPERLY
                                                ;*INSTRUCTION SHOULD MOVE IBUS*4 TO IBUS*5, IBUS*4 IS ALL 1'S
                                                ;*AND IBUS*5 IS ALL 0'S. RESULT SHOULD BE ALL 1'S IN SEL4
                                                BADHEAD
                                                ;******************** TEST 13 ********************

                                        BGNTST
                                        T13::
                                                MYINT
 (1) 013562 013701 002716                       MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
                                                MSTCLR
 (1) 013566 004537 003142                       JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
013572 012761 000377 000004                     MOV     #377,4(R1)      ;PORT4 HI BYTE=1'S
013600 012711 001000                            MOV     #BIT9,(R1)              ;SET ROMI
013604 012761 121105 000006                     MOV     #121105,6(R1)   ;INSTR TO PORT 6.
013612 052711 001400                            BIS     #BIT8!BIT9,(R1) ;CLK INSTR.
013616 000240                                   NOP
013620 012702 177777                            MOV     #-1,R2          ;EXPECT ALL ONES.
013624 116104 000004                            MOVB    4(R1),R4        ;READ FOUND.
013630 020204                                   CMP     R2,R4   ;DATA CORRECT?
013632 001413                                   BEQ     1$
013634                                          ERROR   28
 (5) 013652 104455                              TRAP    C$ERDF
 (6) 013654 000034                              .WORD   28
 (6) 013656 005540                              .WORD   EM28
 (6) 013660 007712                              .WORD   ERR28

013662                                  1$:     ESCAPE  TST
 (3) 013662 104410                              TRAP    C$ESCAPE
 (3) 013664 000002                              .WORD   L10066-.

013666                                  ENDTST
 (3) 013666                             L10066:
 (3) 013666 104401                              TRAP    C$ETST

013670                                          BADHEAD
                                                ;******************** TEST 14 ********************
                                                ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
                                                ;*FLOAT A 1 THOUGH IBUS* REGISTER 0
                                                ;*FLOAT A 0 THROUGH IBUS* REGISTER 0
013670                                          BADHEAD
                                                ;******************** TEST 14 ********************

013670                                  BGNTST
 (3) 013670                             T14::
013670                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1) 013670 004537 003142                       JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
013674 012737 000000 002624                     MOV     #0,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
013702 012705 000001                            MOV     #1,R5           ;START WITH BIT 0

013706                                          MYINT
```

```
  (1)  013706  013701  002716            MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6275   013712                             BGNSEG
  (3)  013712  104404                     TRAP    C$BSEG
6276   013714                     64$:
6277   013714  010561  000004            MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
6278   013720                             ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  013720  004537  003230            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6279   013724  121100                     121100                  ;MOV DATA TO IBUS* REGISTER 0
6280   013726                             ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  013726  004537  003230            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6281   013732  121005                     121005                  ;READ FROM IBU* REGITER 0
6282   013734  116104  000005            MOVB    5(R1),R4        ;PUT ''FOUND'' INTO R4
6283   013740  120504                     CMPB    R5,R4           ;DATA CORRECT?
6284   013742  001414                     BEQ     65$             ;BR IF YES
6285   013744                             BERROR  27              ;ERROR
  (5)  013764  104455                     TRAP    C$ERDF
  (6)  013766  000033                     .WORD   27
  (6)  013770  005511                     .WORD   EM27
  (6)  013772  007630                     .WORD   ERR27
6286   013774                     65$:    ESCAPE  SEG
  (3)  013774  104410                     TRAP    C$ESCAPE
  (3)  013776  000010                     .WORD   10000$-.
6287   014000  000241                     CLC                     ;CLEAR CARRY
6288   014002  106105                     ROLB    R5              ;SHIFT BIT IN R5
6289   014004  001343                     BNE     64$             ;IF R2=0 THEN DONE
6290   014006                             ENDSEG
  (3)  014006                     10000$:
  (3)  014006  104405                     TRAP    C$ESEG
6291   014010  012705  000001            MOV     #1,R5           ;START WITH BIT 0
6292                             ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
6293   014014                             BGNSEG
  (3)  014014  104404                     TRAP    C$BSEG
6294   014016                     67$:
6295   014016  005105                     COM     R5
6296   014020  010561  000004            MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
6297   014024                             ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  014024  004537  003230            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6298   014030  121100                     121100                  ;MOV DATA TO IBUS* REGISTER 0
6299   014032                             ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  014032  004537  003230            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6300   014036  121005                     121005                  ;READ FROM IBUS* REGISTER 0
6301   014040  116104  000005            MOVB    5(R1),R4        ;PUT ''FOUND'' INTO R4
6302   014044  120504                     CMPB    R5,R4           ;DATA CORRECT?
6303   014046  001414                     BEQ     68$             ;BR IF YES
6304   014050                             BERROR  27              ;ERROR
  (5)  014070  104455                     TRAP    C$ERDF
  (6)  014072  000033                     .WORD   27
  (6)  014074  005511                     .WORD   EM27
  (6)  014076  007630                     .WORD   ERR27
6305   014100                     68$:    ESCAPE  SEG
  (3)  014100  104410                     TRAP    C$ESCAPE
  (3)  014102  000012                     .WORD   10001$-.
6306   014104  005105                     COM     R5              ;CHANGE TO FLOATING 1
6307   014106  000241                     CLC                     ;CLEAR CARRY
6308   014110  106105                     ROLB    R5              ;SHIFT BIT IN R5
6309   014112  001341                     BNE     67$             ;IF R2=0 THEN DONE
```

```
 6310  014114                                  ENDSEG
  (3)  014114                          10001$:
  (3)  014114  104405                         TRAP    C$ESEG
 6311  014116                          ENDTST
  (3)  014116                          L10067:
  (3)  014116  104401                         TRAP    C$ETST
 6312
 6313  014120                                  BADHEAD
  (2)                                         ;******************** TEST 15 *********************
 6314                                         ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
 6315                                         ;*FLOAT A 1 THOUGH IBUS* REGISTER 2
 6316                                         ;*FLOAT A 0 THROUGH IBUS* REGISTER 2
 6317  014120                                  BADHEAD
  (2)                                         ;******************** TEST 15 *********************
 6318
 6319  014120                          BGNTST
  (3)  014120                          T15::
 6320  014120                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  014120  004537  003142                 JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 6321  014124  012737  000002  002624         MOV     #2,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
 6322  014132  012705  000001                 MOV     #1,R5           ;START WITH BIT 0
 6323  014136                                  MYINT
  (1)  014136  013701  002716                 MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6324  014142                                  BGNSEG
  (3)  014142  104404                         TRAP    C$BSEG
 6325  014144                          64$:
 6326  014144  010561  000004                 MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 6327  014150                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  014150  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6328  014154  121102                         121100.2                ;MOV DATA TO IBUS* REGISTER 0
 6329  014156                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  014156  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6330  014162  121045                         121005!<2*20>           ;READ FROM IBUS* REGISTER 2
 6331  014164  116104  000005                 MOVB    5(R1),R4        ;PUT ''FOUND'' INTO R4
 6332  014170  120504                         CMPB    R5,R4           ;DATA CORRECT?
 6333  014172  001414                         BEQ     65$             ;BR IF YES
 6334  014174                                  BERROR  27              ;ERROR
  (5)  014214  104455                         TRAP    C$ERDF
  (6)  014216  000033                         .WORD   27
  (6)  014220  005511                         .WORD   EM27
  (6)  014222  007630                         .WORD   ERR27
 6335  014224                          65$:    ESCAPE  SEG
  (3)  014224  104410                         TRAP    C$ESCAPE
  (3)  014226  000010                         .WORD   10000$-.
 6336  014230  000241                         CLC                     ;CLEAR CARRY
 6337  014232  106105                         ROLB    R5              ;SHIFT BIT IN R2
 6338  014234  001343                         BNE     64$             ;IF R2=0 THEN DONE
 6339  014236                                  ENDSEG
  (3)  014236                          10000$:
  (3)  014236  104405                         TRAP    C$ESEG
 6340  014240  012705  000001                 MOV     #1,R5           ;START WITH BIT 0
 6341                                ;69$:    COM     R5              ;CHANGE TO FLOATING ZERO
 6342  014244                                  BGNSEG
  (3)  014244  104404                         TRAP    C$BSEG
 6343  014246                          67$:
 6344  014246  005105                         COM     R5
```

```
 6345  014250  010561  000004                 MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 6346  014254                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  014254  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6347  014260  121102                         121100!2                ;MOV DATA TO IBUS* REGISTER 2
 6348  014262                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  014262  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6349  014266  121045                         121005!<2*20>           ;READ FROM IBUS* REGISTER 2
 6350  014270  116104  000005                 MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
 6351  014274  120504                         CMPB    R5,R4           ;DATA CORRECT?
 6352  014276  001414                         BEQ     68$             ;BR IF YES
 6353  014300                                 BERROR  27              ;ERROR
  (5)  014320  104455                         TRAP    C$ERDF
  (6)  014322  000033                         .WORD   27
  (6)  014324  005511                         .WORD   EM27
  (6)  014326  007630                         .WORD   ERR27
 6354  014330                         68$:    ESCAPE  SEG
  (3)  014330  104410                         TRAP    C$ESCAPE
  (3)  014332  000012                         .WORD   10001$-.
 6355  014334  005105                         COM     R5              ;CHANGE TO FLOATING 1
 6356  014336  000241                         CLC                     ;CLEAR CARRY
 6357  014340  106105                         ROLB    R5              ;SHIFT BIT IN R2
 6358  014342  001341                         BNE     67$             ;IF R2=0 THEN DONE
 6359  014344                                 ENDSEG
  (3)  014344                         10001$:
  (3)  014344  104405                         TRAP    C$ESEG
 6360  014346                         ENDTST
  (3)  014346                         L10070:
  (3)  014346  104401                         TRAP    C$ETST
 6361
 6362  014350                                 BADHEAD
  (2)                                        ;******************** TEST 16 ********************
 6363                                        ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
 6364                                        ;*FLOAT A 1 THOUGH IBUS* REGISTER 4
 6365                                        ;*FLOAT A 0 THROUGH IBUS* REGISTER 4
 6366  014350                                 BADHEAD
  (2)                                        ;******************** TEST 16 ********************
 6367
 6368  014350                         BGNTST
  (3)  014350                         T16::
 6369  014350                                 MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  014350  004537  003142                 JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 6370  014354  012737  000004  002624         MOV     #4,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
 6371  014362  012705  000001                 MOV     #1,R5           ;START WITH BIT 0
 6372  014366                                 MYINT
  (1)  014366  013701  002716                 MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6373  014372                                 BGNSEG
  (3)  014372  104404                         TRAP    C$BSEG
 6374  014374                         64$:
 6375  014374  010561  000004                 MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 6376  014400                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  014400  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6377  014404  121104                         121100!4                ;MOV DATA TO IBUS* REGISTER 4
 6378  014406                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  014406  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6379  014412  121105                         121005!<4*20>           ;READ FROM IBUS* REGISTER 4
 6380  014414  116104  000005                 MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
```

```
 6381  014420  120504                        CMPB    R5,R4           ;DATA CORRECT?
 6382  014422  001414                        BEQ     65$             ;BR IF YES
 6383  014424                                BERROR  27              ;ERROR
  (5)  014444  104455                        TRAP    C$ERDF
  (6)  014446  000033                        .WORD   27
  (6)  014450  005511                        .WORD   EM27
  (6)  014452  007630                        .WORD   ERR27
 6384  014454                        65$:    ESCAPE  SEG
  (3)  014454  104410                        TRAP    C$ESCAPE
  (3)  014456  000010                        .WORD   10000$-.
 6385  014460  000241                        CLC                     ;CLEAR CARRY
 6386  014462  106105                        ROLB    R5              ;SHIFT BIT IN R2
 6387  014464  001343                        BNE     64$             ;IF R2=0 THEN DONE
 6388  014466                                ENDSEG
  (3)  014466                        10000$:
  (3)  014466  104405                        TRAP    C$ESEG
 6389  014470  012705  000001                MOV     #1,R5           ;START WITH BIT 0
 6390                             ;69$:      COM     R5              ;CHANGE TO FLOATING ZERO
 6391  014474                                BGNSEG
  (3)  014474  104404                        TRAP    C$BSEG
 6392  014476                        67$:
 6393  014476  005105                        COM     R5
 6394  014500  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 6395  014504                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  014504  004537  003230                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6396  014510  121104                        121100!4                ;MOV DATA TO IBUS* REGISTER 4
 6397  014512                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  014512  004537  003230                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6398  014516  121105                        121005!<4*20>           ;READ FROM IBUS* REGISTER 4
 6399  014520  116104  000005                MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
 6400  014524  120504                        CMPB    R5,R4           ;DATA CORRECT?
 6401  014526  001414                        BEQ     68$             ;BR IF YES
 6402  014530                                BERROR  27              ;ERROR
  (5)  014550  104455                        TRAP    C$ERDF
  (6)  014552  000033                        .WORD   27
  (6)  014554  005511                        .WORD   EM27
  (6)  014556  007630                        .WORD   ERR27
 6403  014560                        68$:    ESCAPE  SEG
  (3)  014560  104410                        TRAP    C$ESCAPE
  (3)  014562  000012                        .WORD   10001$-.
 6404  014564  005105                        COM     R5              ;CHANGE TO FLOATING 1
 6405  014566  000241                        CLC                     ;CLEAR CARRY
 6406  014570  106105                        ROLB    R5              ;SHIFT BIT IN R2
 6407  014572  001341                        BNE     67$             ;IF R2=0 THEN DONE
 6408  014574                                ENDSEG
  (3)  014574                        10001$:
  (3)  014574  104405                        TRAP    C$ESEG
 6409  014576                        ENDTST
  (3)  014576                        L10071:
  (3)  014576  104401                        TRAP    C$ETST
 6410
 6411  014600                                BADHEAD
  (2)                                       ;********************* TEST 17 *********************
 6412                                       ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
 6413                                       ;*FLOAT A 1 THOUGH IBUS* REGISTER 5
 6414                                       ;*FLOAT A 0 THROUGH IBUS* REGISTER 5
```

```
 6415  014600                                          BADHEAD
  (2)                                                ;****************** TEST 17 ******************
 6416
 6417  014600                                    BGNTST
  (3)  014600                                    T17::
 6418  014600                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  014600  004537  003142                          JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 6419  014604  012737  000005  002624                  MOV     #5,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
 6420  014612  012705  000001                          MOV     #1,R5           ;START WITH BIT 0
 6421  014616                                          MYINT
  (1)  014616  013701  002716                          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6422  014622                                          BGNSEG
  (3)  014622  104404                                  TRAP    C$BSEG
 6423  014624                                    64$:
 6424  014624  010561  000004                          MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 6425  014630                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  014630  004537  003230                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6426  014634  121105                                  121100!5                ;MOV DATA TO IBUS* REGISTER 5
 6427  014636                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  014636  004537  003230                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6428  014642  121125                                  121005!<5*20>           ;READ FROM IBUS* REGISTER 5
 6429  014644  116104  000005                          MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
 6430  014650  120504                                  CMPB    R5,R4           ;DATA CORRECT?
 6431  014652  001414                                  BEQ     65$             ;BR IF YES
 6432  014654                                          BERROR  27              ;ERROR
  (5)  014674  104455                                  TRAP    C$ERDF
  (6)  014676  000033                                  .WORD   27
  (6)  014700  005511                                  .WORD   EM27
  (6)  014702  007630                                  .WORD   ERR27
 6433  014704                                    65$:  ESCAPE  SEG
  (3)  014704  104410                                  TRAP    C$ESCAPE
  (3)  014706  000010                                  .WORD   10000$-.
 6434  014710  000241                                  CLC                     ;CLEAR CARRY
 6435  014712  106105                                  ROLB    R5              ;SHIFT BIT IN R5
 6436  014714  001343                                  BNE     64$             ;IF R5=0 THEN DONE
 6437  014716                                          ENDSEG
  (3)  014716                                    10000$:
  (3)  014716  104405                                  TRAP    C$ESEG
 6438  014720  012705  000001                          MOV     #1,R5           ;START WITH BIT 0
 6439                                           ;69$:  COM     R5              ;CHANGE TO FLOATING ZERO
 6440  014724                                          BGNSEG
  (3)  014724  104404                                  TRAP    C$BSEG
 6441  014726                                    67$:
 6442  014726  005105                                  COM     R5
 6443  014730  010561  000004                          MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 6444  014734                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  014734  004537  003230                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6445  014740  121105                                  121100!5                ;MOV DATA TO IBUS* REGISTER 5
 6446  014742                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  014742  004537  003230                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6447  014746  121125                                  121005!<5*20>           ;READ FROM IBUS* REGISTER 5
 6448  014750  116104  000005                          MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
 6449  014754  120504                                  CMPB    R5,R4           ;DATA CORRECT?
 6450  014756  001414                                  BEQ     68$             ;BR IF YES
 6451  014760                                          BERROR  27              ;ERROR
  (5)  015000  104455                                  TRAP    C$ERDF
```

```
  (6)  015002  000033                    .WORD   27
  (6)  015004  005511                    .WORD   EM27
  (6)  015006  007630                    .WORD   ERR27
6452   015010                  68$:      ESCAPE  SEG
  (3)  015010  104410                    TRAP    C$ESCAPE
  (3)  015012  000012                    .WORD   10001$-.
6453   015014  005105                    COM     R5              ;CHANGE TO FLOATING 1
6454   015016  000241                    CLC                     ;CLEAR CARRY
6455   015020  106105                    ROLB    R5              ;SHIFT BIT IN R5
6456   015022  001341                    BNE     67$             ;IF R5=0 THEN DONE
6457   015024                            ENDSEG
  (3)  015024                  10001$:
  (3)  015024  104405                    TRAP    C$ESEG
6458   015026                  ENDTST
  (3)  015026                  L10072:
  (3)  015026  104401                    TRAP    C$ETST
6459
6460   015030                            BADHEAD
  (2)                                     ;********** TEST 18 **********
6461                                      ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
6462                                      ;*FLOAT A 1 THOUGH IBUS* REGISTER 10
6463                                      ;*FLOAT A 0 THROUGH IBUS* REGISTER 10
6464   015030                            BADHEAD
  (2)                                     ;********** TEST 18 **********
6465
6466   015030                  BGNTST
  (3)  015030                  T18::
6467   015030                            MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  015030  004537  003142              JSR     R5,.MSTCLR             ;CLEAR M8200,4,7
6468   015034  012737  000010  002624      MOV     #10,MR0         ;SAVE REGISTER ADDRESS FOR TYPEOUT
6469   015042  012705  000001              MOV     #1,R5           ;START WITH BIT 0
6470   015046                            MYINT
  (1)  015046  013701  002716              MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6471   015052                            BGNSEG
  (3)  015052  104404                    TRAP    C$BSEG
6472   015054                  64$:
6473   015054  010561  000004              MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
6474   015060  042761  000141  000004      BIC     #141,4(R1)      ;CLEAR UNWANTED BITS
6475   015066                            ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  015066  004537  003230              JSR     R5,.ROMCLK             ;CLOCK INSTRUCTION
6476   015072  121110                    121100!10               ;MOV DATA TO IBUS* REGISTER 10
6477   015074                            ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  015074  004537  003230              JSR     R5,.ROMCLK             ;CLOCK INSTRUCTION
6478   015100  121205                    121005!<10*20>          ;READ FROM IBUS* REGISTER 10
6479   015102  010502                    MOV     R5,R2
6480   015104  042705  000141            BIC     #141,R5         ;CLEAR UNWANTED BITS
6481   015110  116104  000005            MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
6482   015114  120504                    CMPB    R5,R4           ;DATA CORRECT?
6483   015116  001414                    BEQ     65$             ;BR IF YES
6484   015120                            BERROR  27              ;ERROR
  (5)  015140  104455                    TRAP    C$ERDF
  (6)  015142  000033                    .WORD   27
  (6)  015144  005511                    .WORD   EM27
  (6)  015146  007630                    .WORD   ERR27
6485   015150                  65$:      ESCAPE  SEG
  (3)  015150  104410                    TRAP    C$ESCAPE
```

```
  (3)  015152  000012                        .WORD   10000$-.
6486   015154  010205                        MOV     R2,R5
6487   015156  000241                        CLC                     ;CLEAR CARRY
6488   015160  106105                        ROLB    R5              ;SHIFT BIT IN R5
6489   015162  001334                        BNE     64$             ;IF R5=0 THEN DONE
6490   015164                                ENDSEG
  (3)  015164                          10000$:
  (3)  015164  104405                        TRAP    C$ESEG
6491   015166  012705  000001                MOV     #1,R5           ;START WITH BIT 0
6492                                 ;69$:    COM     R5              ;CHANGE TO FLOATING ZERO
6493   015172                                BGNSEG
  (3)  015172  104404                        TRAP    C$BSEG
6494   015174                          67$:
6495   015174  005105                        COM     R5
6496   015176  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
6497   015202  042761  000141  000004        BIC     #141,4(R1)      ;CLEAR UNWANTED BITS
6498   015210                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  015210  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6499   015214  121110                        121100!10               ;MOV DATA TO IBUS* REGISTER 10
6500   015216                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  015216  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6501   015222  121205                        121005!<10*20>          ;READ FROM IBUS* REGISTER 10
6502   015224  010502                        MOV     R5,R2
6503   015226  042705  000141                BIC     #141,R5         ;CLEAR UNWANTED BITS
6504   015232  116104  000005                MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
6505   015236  120504                        CMPB    R5,R4           ;DATA CORRECT?
6506   015240  001414                        BEQ     68$             ;BR IF YES
6507   015242                                BERROR  27              ;ERROR
  (5)  015262  104455                        TRAP    C$ERDF
  (6)  015264  000033                        .WORD   27
  (6)  015266  005511                        .WORD   EM27
  (6)  015270  007630                        .WORD   ERR27
6508   015272                          68$:  ESCAPE  SEG
  (3)  015272  104410                        TRAP    C$ESCAPE
  (3)  015274  000014                        .WORD   10001$-.
6509   015276  010205                        MOV     R2,R5
6510   015300  005105                        COM     R5              ;CHANGE TO FLOATING 1
6511   015302  000241                        CLC                     ;CLEAR CARRY
6512   015304  106105                        ROLB    R5              ;SHIFT BIT IN R5
6513   015306  001332                        BNE     67$             ;IF R5=0 THEN DONE
6514   015310                                ENDSEG
  (3)  015310                          10001$:
  (3)  015310  104405                        TRAP    C$ESEG
6515   015312                          ENDTST
  (3)  015312                          L10073:
  (3)  015312  104401                        TRAP    C$ETST
6516
6517   015314                                BADHEAD
  (2)
6518                                          ;******************** TEST 19 ********************
6519                                          ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
6520                                          ;*FLOAT A 1 THOUGH IBUS* REGISTER 11
6521   015314                                 ;*FLOAT A 0 THROUGH IBUS* REGISTER 11
  (2)                                         BADHEAD
6522                                          ;******************** TEST 19 ********************
6523   015314                          BGNTST
```

```
  (3)  015314                                   T19::
6524   015314
  (1)  015314  004537  003142                            MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  015314  004537  003142                            JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
6525   015320  012737  000011  002624                    MOV     #11,MR0         ;SAVE REGISTER ADDRESS FOR TYPEOUT
6526   015326  012705  000001                            MOV     #1,R5           ;START WITH BIT 0
6527   015332                                            MYINT
  (1)  015332  013701  002716                            MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6528   015336                                            BGNSEG
  (3)  015336  104404                                    TRAP    C$BSEG
6529   015340                                64$:
6530   015340  010561  000004                            MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
6531   015344  042761  000262  000004                    BIC     #262,4(R1)      ;CLEAR UNWANTED BITS
6532   015352                                            ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  015352  004537  003230                            JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6533   015356  121111                                    121100!11               ;MOV DATA TO IBUS* REGISTER 11
6534   015360                                            ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  015360  004537  003230                            JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6535   015364  121225                                    121005!<11*20>          ;READ FROM IBUS* REGISTER 11
6536   015366  010502                                    MOV     R5,R2
6537   015370  042705  000262                            BIC     #262,R5         ;CLEAR UNWANTED BITS
6538   015374  116104  000005                            MOVB    5(R1),R4        ;PUT 'FOUND'' INTO R4
6539   015400  042704  000020                            BIC     #20,R4
6540   015404  120504                                    CMPB    R5,R4           ;DATA CORRECT?
6541   015406  001414                                    BEQ     65$             ;BR IF YES
6542   015410                                            BERROR  27              ;ERROR
  (5)  015430  104455                                    TRAP    C$ERDF
  (6)  015432  000033                                    .WORD   27
  (6)  015434  005511                                    .WORD   EM27
  (6)  015436  007630                                    .WORD   ERR27
```

```
 6544  015440                          65$:    ESCAPE  SEG
  (3)  015440  104410                           TRAP    C$ESCAPE
  (3)  015442  000012                           .WORD   10000$-.
 6545  015444  010205                           MOV     R2,R5
 6546  015446  000241                           CLC                     ;CLEAR CARRY
 6547  015450  106105                           ROLB    R5              ;SHIFT BIT IN R5
 6548  015452  001332                           BNE     64$             ;IF R5=0 THEN DONE
 6549  015454                                   ENDSEG
  (3)  015454                          10000$:
  (3)  015454  104405                           TRAP    C$ESEG
 6550  015456  012705  000001                   MOV     #1,R5           ;START WITH BIT 0
 6551                               ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
 6552  015462                                   BGNSEG
  (3)  015462  104404                           TRAP    C$BSEG
 6553  015464                          67$:
 6554  015464  005105                           COM     R5
 6555  015466  010561  000004                   MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 6556  015472  042761  000262  000004           BIC     #262,4(R1)      ;CLEAR UNWANTED BITS
 6557  015500                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  015500  004537  003230                   JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6558  015504  121111                           121100!11               ;MOV DATA TO IBUS* REGISTER 11
 6559  015506                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  015506  004537  003230                   JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6560  015512  121225                           121005!<11*20>          ;READ FROM IBUS* REGISTER 11
```

```
 6562  015514  010502                        MOV     R5,R2
 6563  015516  042705  000262                BIC     #262,R5         ;CLEAR UNWANTED BITS
 6564  015522  052705  000020                BIS     #20,R5          ;ADD THESE BITS
 6565  015526  116104  000005.               MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
 6566  015532  120504                        CMPB    R5,R4           ;DATA CORRECT?
 6567  015534  001414                        BEQ     68$             ;BR IF YES
 6568  015536                                BERROR  27              ;ERROR
   (5) 015556  104455                        TRAP    C$ERDF
   (6) 015560  000033                        .WORD   27
   (6) 015562  005511                        .WORD   EM27
   (6) 015564  007630                        .WORD   ERR27
 6569  015566                        68$:    ESCAPE  SEG
   (3) 015566  104410                        TRAP    C$ESCAPE
   (3) 015570  000014                        .WORD   10001$-.
 6570  015572  010205                        MOV     R2,R5
 6571  015574  005105                        COM     R5              ;CHANGE TO FLOATING 1
 6572  015576  000241                        CLC                     ;CLEAR CARRY
 6573  015600  106105                        ROLB    R5              ;SHIFT BIT IN R5
 6574  015602  001330                        BNE     67$             ;IF R5=0 THEN DONE
 6575  015604                                ENDSEG
   (3) 015604                        10001$:
   (3) 015604  104405                        TRAP    C$ESEG
 6576  015606                        ENDIST
   (3) 015606                        L10074:
   (3) 015606  104401                        TRAP    C$ETST
 6577
 6578  015610                                BADHEAD
   (2)                                     ;****************** TEST 20 ******************
 6579                                      ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
 6580                                      ;*FLOAT A 1 THOUGH IBUS REGISTER 0
 6581                                      ;*FLOAT A 0 THROUGH IBUS REGISTER 0
 6582  015610                                BADHEAD
   (2)                                     ;****************** TEST 20 ******************
 6583
 6584  015610                        BGNTST
   (3) 015610                        T20::
 6585  015610                                MSTCLR                  ;MASTER CLEAR M8200,4,7
   (1) 015610  004537  003142                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 6586  015614  012737  000000  002624        MOV     #0,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
 6587  015622  012705  000001                MOV     #1,R5           ;START WITH BIT 0
 6588  015626                                MYINT
   (1) 015626  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6589  015632                                BGNSEG
   (3) 015632  104404                        TRAP    C$BSEG
 6590  015634                        64$:
 6591  015634  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 6592  015640                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
   (1) 015640  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6593  015644  122100                        122100                  ;MOV DATA TO IBUS* REGISTER 0
 6594  015646                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
   (1) 015646  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6595  015652  021005                        21005                   ;READ FROM IBUS* REGISTER 0
 6596  015654  116104  000005                MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
 6597  015660  120504                        CMPB    R5,R4           ;DATA CORRECT?
 6598  015662  001414                        BEQ     65$             ;BR IF YES
 6599  015664                                BERROR  29              ;ERROR
```

```
   (5)  015704  104455                    TRAP    C$ERDF
   (6)  015706  000035                    .WORD   29
   (6)  015710  005571                    .WORD   EM29
   (6)  015712  007770                    .WORD   ERR29
 6600   015714                  65$:      ESCAPE  SEG
   (3)  015714  104410                    TRAP    C$ESCAPE
   (3)  015716  000010                    .WORD   10000$-.
 6601   015720  000241                    CLC                     ;CLEAR CARRY
 6602   015722  106105                    ROLB    R5              ;SHIFT BIT IN R5
 6603   015724  001343                    BNE     64$             ;IF R5=0 THEN DONE
 6604   015726                            ENDSEG
   (3)  015726                  10000$:
   (3)  015726  104405                    TRAP    C$ESEG
 6605   015730  012705  000001            MOV     #1,R5           ;START WITH BIT 0
 6606                          ;69$:       COM     R5              ;CHANGE TO FLOATING ZERO
 6607   015734                            BGNSEG
   (3)  015734  104404                    TRAP    C$BSEG
 6608   015736                  67$:
 6609   015736  005105                    COM     R5
 6610   015740  010561  000004            MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 6611   015744                            ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
   (1)  015744  004537  003230            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6612   015750  122100                    122100                  ;MOV DATA TO IBUS* REGISTER 0
 6613   015752                            ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
   (1)  015752  004537  003230            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6614   015756  021005                    21005                   ;READ FROM IBUS* REGISTER 0
 6615   015760  116104  000005            MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 6616   015764  120504                    CMPB    R5,R4           ;DATA CORRECT?
 6617   015766  001414                    BEQ     68$             ;BR IF YES
 6618   015770                            BERROR  29              ;ERROR
   (5)  016010  104455                    TRAP    C$ERDF
   (6)  016012  000035                    .WORD   29
   (6)  016014  005571                    .WORD   EM29
   (6)  016016  007770                    .WORD   ERR29
 6619   016020                  68$:      ESCAPE  SEG
   (3)  016020  104410                    TRAP    C$ESCAPE
   (3)  016022  000012                    .WORD   10001$-.
 6620   016024  005105                    COM     R5              ;CHANGE TO FLOATING 1
 6621   016026  000241                    CLC                     ;CLEAR CARRY
 6622   016030  106105                    ROLB    R5              ;SHIFT BIT IN P5
 6623   016032  001341                    BNE     67$             ;IF R5=0 THEN DONE
 6624   016034                            ENDSEG
   (3)  016034                  10001$:
   (3)  016034  104405                    TRAP    C$ESEG
 6625   016036                  ENDTST
   (3)  016036                  L10075:
   (3)  016036  104401                    TRAP    C$ETST
 6626
 6627   016040                            BADHEAD
   (2)                                    ;******************** TEST 21 ********************
 6628                                     ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
 6629                                     ;*FLOAT A 1 THOUGH IBUS REGISTER 1
 6630                                     ;*FLOAT A 0 THROUGH IBUS REGISTER 1
 6631   016040                            BADHEAD
   (2)                                    ;******************** TEST 21 ********************
 6632
```

```
 6633  016040                                 BGNTST
  (3)  016040                                 T21::
 6634  016040                                       MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  016040  004537  003142                       JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 6635  016044  012737  000001  002624               MOV     #1,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
 6636  016052  012705  000001                       MOV     #1,R5           ;START WITH BIT 0
 6637  016056                                       MYINT
  (1)  016056  013701  002716                       MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6638  016062                                       BGNSEG
  (3)  016062  104404                               TRAP    C$BSEG
 6639  016064                                 64$:
 6640  016064  010561  000004                       MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 6641  016070                                       ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  016070  004537  003230                       JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6642  016074  122101                               122100!1                ;MOV DATA TO IBUS* REGISTER 1
 6643  016076                                       ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  016076  004537  003230                       JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6644  016102  021025                               21005!<1*20>            ;READ FROM IBUS* REGISTER 1
 6645  016104  116104  000005                       MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
 6646  016110  120504                               CMPB    R5,R4           ;DATA CORRECT?
 6647  016112  001414                               BEQ     65$             ;BR IF YES
 6648  016114                                       BERROR  29              ;ERROR
  (5)  016134  104455                               TRAP    C$ERDF
  (6)  016136  000035                               .WORD   29
  (6)  016140  005571                               .WORD   EM29
  (6)  016142  007770                               .WORD   ERR29
 6649  016144                                 65$:  ESCAPE  SEG
  (3)  016144  104410                               TRAP    C$ESCAPE
  (3)  016146  000010                               .WORD   10000$-.
 6650  016150  000241                               CLC                     ;CLEAR CARRY
 6651  016152  106105                               ROLB    R5              ;SHIFT BIT IN R5
 6652  016154  001343                               BNE     64$             ;IF R5=0 THEN DONE
 6653  016156                                       ENDSEG
  (3)  016156                                 10000$:
  (3)  016156  104405                               TRAP    C$ESEG
 6654  016160  012705  000001                       MOV     #1,R5           ;START WITH BIT 0
 6655                                       ;69$:  COM     R5              ;CHANGE TO FLOATING ZERO
 6656  016164                                       BGNSEG
  (3)  016164  104404                               TRAP    C$BSEG
 6657  016166                                 67$:
 6658  016166  005105                               COM     R5
 6659  016170  010561  000004                       MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 6660  016174                                       ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  016174  004537  003230                       JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6661  016200  122101                               122100!1                ;MOV DATA TO IBUS* REGISTER 1
 6662  016202                                       ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  016202  004537  003230                       JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6663  016206  021025                               21005!<1*20>            ;READ FROM IBUS* REGISTER 1
 6664  016210  116104  000005                       MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
 6665  016214  120504                               CMPB    R5,R4           ;DATA CORRECT?
 6666  016216  001414                               BEQ     68$             ;BR IF YES
 6667  016220                                       BERROR  29              ;ERROR
  (5)  016240  104455                               TRAP    C$ERDF
  (6)  016242  000035                               .WORD   29
  (6)  016244  005571                               .WORD   EM29
  (6)  016246  007770                               .WORD   ERR29
```

```
 6668  016250                          68$:    ESCAPE  SEG
  (3)  016250  104410                         TRAP    C$ESCAPE
  (3)  016252  000012                         .WORD   10001$-.
 6669  016254  005105                         COM     R5              ;CHANGE TO FLOATING 1
 6670  016256  000241                         CLC                     ;CLEAR CARRY
 6671  016260  106105                         ROLB    R5              ;SHIFT BIT IN R5
 6672  016262  001341                         BNE     67$             ;IF R5=0 THEN DONE
 6673  016264                                 ENDSEG
  (3)  016264                          10001$:
  (3)  016264  104405                         TRAP    C$ESEG
 6674  016266                          ENDTST
  (3)  016266                          L10076:
  (3)  016266  104401                         TRAP    C$ETST
 6675
 6676  016270                                 BADHEAD
  (2)                                        ;******************** TEST 22 ********************
 6677                                        ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
 6678                                        ;*FLOAT A 1 THOUGH IBUS REGISTER 2
 6679                                        ;*FLOAT A 0 THROUGH IBUS REGISTER 2
 6680  016270                                 BADHEAD
  (2)                                        ;******************** TEST 22 ********************
 6681
 6682  016270                          BGNTST
  (3)  016270                          T22::
 6683  016270                                 MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  016270  004537  003142                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 6684  016274  012737  000002  002624         MOV     #2,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
 6685  016302  012705  000001                MOV     #1,R5           ;START WITH BIT 0
 6686  016306                                 MYINT
  (1)  016306  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6687  016312                                 BGNSEG
  (3)  016312  104404                        .RAP    C$BSEG
 6688  016314                          64$:
 6689  016314  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 6690  016320                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  016320  004537  003230                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6691  016324  122102                        122100!2                ;MOV DATA TO IBUS* REGISTER 2
 6692  016326                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  016326  004537  003230                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6693  016332  021045                        21005!<2*20>            ;READ FROM IBUS* REGISTER 2
 6694  016334  116104  000005                MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
 6695  016340  120504                        CMPB    R5,R4           ;DATA CORRECT?
 6696  016342  001414                        BEQ     65$             ;BR IF YES
 6697  016344                                 BERROR  29              ;ERROR
  (5)  016364  104455                        TRAP    C$ERDF
  (6)  016366  000035                        .WORD   29
  (6)  016370  005571                        .WORD   EM29
  (6)  016372  007770                        .WORD   ERR29
 6698  016374                          65$:   ESCAPE  SEG
  (3)  016374  104410                        TRAP    C$ESCAPE
  (3)  016376  000010                        .WORD   10000$-.
 6699  016400  000241                        CLC                     ;CLEAR CARRY
 6700  016402  106105                        ROLB    R5              ;SHIFT BIT IN R5
 6701  016404  001343                        BNE     64$             ;IF R5=0 THEN DONE
 6702  016406                                 ENDSEG
  (3)  016406                          10000$:
```

```
  (3)  016406  104405
6703   016410  012705  000001          TRAP    C$ESEG
6704                                   MOV     #1,R5           ;START WITH BIT 0
6705   016414                  ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
  (3)  016414  10440-                  BGNSEG
6706   016416                          TRAP    C$BSEG
6707   016416  005105          67$:
                                       COM     R5
6708   016420  010561  000004          MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
6709   016424                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  016424  004537  003230          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6710   016430  122102                  122100!2                ;MOV DATA TO IBUS* REGISTER 2
6711   016432                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  016432  004537  003230          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6712   016436  021045                  21005!<2*20>            ;READ FROM IBUS* REGISTER 2
6713   016440  116104  000005          MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
6714   016444  120504                  CMPB    R5,R4           ;DATA CORRECT?
6715   016446  001414                  BEQ     68$             ;BR IF YES
6716   016450                          BERROR  29              ;ERROR
  (5)  016470  104455                  TRAP    C$ERDF
  (6)  016472  000035                  .WORD   29
  (6)  016474  005571                  .WORD   EM29
  (6)  016476  007770                  .WORD   ERR29
6717   016500                  68$:    ESCAPE  SEG
  (3)  016500  104410                  TRAP    C$ESCAPE
  (3)  016502  000012                  .WORD   10001$-.
6718   016504  005105                  COM     R5              ;CHANGE TO FLOATING 1
6719   016506  000241                  CLC                     ;CLEAR CARRY
6720   016510  106105                  RULB    R5              ;SHIFT BIT IN R5
6721   016512  001341                  BNE     67$             ;IF R5=0 THEN DONE
6722   016514                          ENDSEG
  (3)  016514                  10001$:
  (3)  016514  104405                  TRAP    C$ESEG
6723   016516                  ENDTST
  (3)  016516                  L10077:
  (3)  016516  104401                  TRAP    C$ETST
6724
6725   016520                          BADHEAD
  (2)                                  ;******************** TEST 23 ********************
6726                                   ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
6727                                   ;*FLOAT A 1 THOUGH IBUS REGISTER 3
6728                                   ;*FLOAT A 0 THROUGH IBUS REGISTER 3
6729   016520                          BADHEAD
  (2)                                  ;******************** TEST 23 ********************
6730
6731   016520                  BGNTST
  (3)  016520                  T23::
6732   016520                          MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  016520  004537  003142          JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
6733   016524  012737  000003  002624  MOV     #3,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
6734   016532  012705  000001          MOV     #1,R5           ;START WITH BIT 0
6735   016536                          MYINT
  (1)  016536  013701  002716          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6736   016542                          BGNSEG
  (3)  016542  104404                  TRAP    C$BSEG
6737   016544                  64$:
6738   016544  010561  000004          MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
```

```
6739  016550
 (1)  016550  004537  003230                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
                                            JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6740  016554  122103                        122100!3                ;MOV DATA TO IBUS* REGISTER 3
6741  016556                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  016556  004537  003230                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6742  016562  021065                        21005!<3*20>            ;READ FROM IBUS* REGISTER 3
6743  016564  116104  000005                MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
6744  016570  120504                        CMPB    R5,R4           ;DATA CORRECT?
6745  016572  001414                        BEQ     65$             ;BR IF YES
6746  016574                                BERROR  29              ;ERROR
 (5)  016614  104455                        TRAP    C$ERDF
 (6)  016616  000035                        .WORD   29
 (6)  016620  005571                        .WORD   EM29
 (6)  016622  007770                        .WORD   ERR29
6747  016624                        65$:    ESCAPE  SEG
 (3)  016624  104410                        TRAP    C$ESCAPE
 (3)  016626  000010                        .WORD   10000$-.
6748  016630  000241                        CLC                     ;CLEAR CARRY
6749  016632  106105                        ROLB    R5              ;SHIFT BIT IN R5
6750  016634  001343                        BNE     64$             ;IF R5=0 THEN DONE
6751  016636                                ENDSEG
 (3)  016636                        10000$:
 (3)  016636  104405                        TRAP    C$ESEG
6752  016640  012705  000001                MOV     #1,R5           ;START WITH BIT 0
6753                             ;69$:      COM     R5              ;CHANGE TO FLOATING ZERO
6754  016644                                BGNSEG
 (3)  016644  104404                        TRAP    C$BSEG
6755  016646                        67$:
6756  016646  005105                        COM     R5
6757  016650  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
6758  016654                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  016654  004537  003230                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6759  016660  122103                        122100!3                ;MOV DATA TO IBUS* REGISTER 3
6760  016662                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  016662  004537  003230                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6761  016666  021065                        21005!<3*20>            ;READ FROM IBUS* REGISTER 3
6762  016670  116104  000005                MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
6763  016674  120504                        CMPB    R5,R4           ;DATA CORRECT?
6764  016676  001414                        BEQ     68$             ;BR IF YES
6765  016700                                BERROR  29              ;ERROR
 (5)  016720  104455                        TRAP    C$ERDF
 (6)  016722  000035                        .WORD   29
 (6)  016724  005571                        .WORD   EM29
 (6)  016726  007770                        .WORD   ERR29
6766  016730                        68$:    ESCAPE  SEG
 (3)  016730  104410                        TRAP    C$ESCAPE
 (3)  016732  000012                        .WORD   10001$-.
6767  016734  005105                        COM     R5              ;CHANGE TO FLOATING 1
6768  016736  000241                        CLC                     ;CLEAR CARRY
6769  016740  106105                        ROLB    R5              ;SHIFT BIT IN R5
6770  016742  001341                        BNE     67$             ;IF R5=0 THEN DONE
6771  016744                                ENDSEG
 (3)  016744                        10001$:
 (3)  016744  104405                        TRAP    C$ESEG
6772  016746                        ENDTST
 (3)  016746                        L10100:
```

```
  (3)  016746  104401                          TRAP    C$ETST
6773
6774  016750                                   BADHEAD
  (2)                                          ;******************** TEST 24 ********************
6775                                           ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
6776                                           ;*FLOAT A 1 THOUGH IBUS REGISTER 4
6777                                           ;*FLOAT A 0 THROUGH IBUS REGISTER 4
6778  016750                                   BADHEAD
  (2)                                          ;******************** TEST 24 ********************
6779
6780  016750                          BGNTST
  (3)  016750                          T24::
6781  016750                                   MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  016750  004537  003142                 JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
6782  016754  012737  000004  002624           MOV     #4,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
6783  016762  012705  000001                 MOV     #1,R5           ;START WITH BIT 0
6784  016766                                   MYINT
  (1)  016766  013701  002716                 MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6785  016772                                   BGNSEG
  (3)  016772  104404                          TRAP    C$BSEG
6786  016774                          64$:
6787  016774  010561  000004                 MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
6788  017000                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  017000  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6789  017004  122104                          122100!4                ;MOV DATA TO IBUS* REGISTER 4
6790  017006                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  017006  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6791  017012  021105                          21005!<4*20>            ;READ FROM IBUS* REGISTER 4
6792  017014  116104  000005                 MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
6793  017020  120504                          CMPB    R5,R4           ;DATA CORRECT?
6794  017022  001414                          BEQ     65$             ;BR IF YES
6795  017024                                   BERROR  29              ;ERROR
  (5)  017044  104455                          TRAP    C$ERDF
  (6)  017046  000035                          .WORD   29
  (6)  017050  005571                          .WORD   EM29
  (6)  017052  007770                          .WORD   ERR29
6796  017054                          65$:     ESCAPE  SEG
  (3)  017054  104410                          TRAP    C$ESCAPE
  (3)  017056  000010                          .WORD   10000$-.
6797  017060  000241                          CLC                     ;CLEAR CARRY
6798  017062  106105                          ROLB    R5              ;SHIFT BIT IN R5
6799  017064  001343                          BNE     64$             ;IF R5=0 THEN DONE
6800  017066                                   ENDSEG
  (3)  017066                          10000$:
  (3)  017066  104405                          TRAP    C$ESEG
6801  017070  012705  000001                 MOV     #1,R5           ;START WITH BIT 0
6802                                  ;69$:    COM     R5              ;CHANGE TO FLOATING ZERO
6803  017074                                   BGNSEG
  (3)  017074  104404                          TRAP    C$BSEG
6804  017076                          67$:
6805  017076  005105                          COM     R5
6806  017100  010561  000004                 MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
6807  017104                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  017104  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6808  017110  122104                          122100!4                ;MOV DATA TO IBUS* REGISTER 4
6809  017112                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
```

```
  (1)  017112  004537  003230                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6810   017116  021105                            21005!<4*20>            ;READ FROM IBUS* REGISTER 4
6811   017120  116104  000005                    MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
6812   017124  120504                            CMPB    R5,R4           ;DATA CORRECT?
6813   017126  001414                            BEQ     68$             ;BR IF YES
6814   017130                                    BERROR  29              ;ERROR
  (5)  017150  104455                            TRAP    C$ERDF
  (6)  017152  000035                            .WORD   29
  (6)  017154  005571                            .WORD   EM29
  (6)  017156  007770                            .WORD   ERR29
6815   017160                            68$:    ESCAPE  SEG
  (3)  017160  104410                            TRAP    C$ESCAPE
  (3)  017162  000012                            .WORD   10001$-.
6816   017164  005105                            COM     R5              ;CHANGE TO FLOATING 1
6817   017166  000241                            CLC                     ;CLEAR CARRY
6818   017170  106105                            ROLB    R5              ;SHIFT BIT IN R5
6819   017172  001341                            BNE     67$             ;IF R5=0 THEN DONE
6820   017174                                    ENDSEG
  (3)  017174                            10001$:
  (3)  017174  104405                            TRAP    C$ESEG
6821   017176                            ENDTST
  (3)  017176                            L10101:
  (3)  017176  104401                            TRAP    C$ETST
6822
6823   017200                                    BADHEAD
  (2)    .                                        ;******************** TEST 25 ********************
6824                                              ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
6825                                              ;*FLOAT A 1 THOUGH IBUS REGISTER 5
6826                                              ;*FLOAT A 0 THROUGH IBUS REGISTER 5
6827   017200                                     BADHEAD
  (2)                                             ;******************** TEST 25 ********************
6828
6829   017200                            BGNTST
  (3)  017200                            T25::
6830   017200                                    MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  017200  004537  003142                    JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
6831   017204  012737  000005  002624            MOV     #5,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
6832   017212  012705  000001                    MOV     #1,R5           ;START WITH BIT 0
6833   017216                                    MYINT
  (1)  017216  013701  002716                    MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6834   017222                                    BGNSEG
  (3)  017222  104404                            TRAP    C$BSEG
6835   017224                            64$:    .
6836   017224  010561  000004                    MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
6837   017230                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  017230  004537  003230                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6838   017234  122105                            122100!5                ;MOV DATA TO IBUS* REGISTER 5
6839   017236                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  017236  004537  003230                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6840   017242  021125                            21005!<5*20>            ;READ FROM IBUS* REGISTER 5
6841   017244  116104  000005                    MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
6842   017250  120504                            CMPB    R5,R4           ;DATA CORRECT?
6843   017252  001414                            BEQ     65$             ;BR IF YES
6844   017254                                    BERROR  29              ;ERROR
  (5)  017274  104455                            TRAP    C$ERDF
  (6)  017276  000035                            .WORD   29
```

```
  (6)  017300  005571                    .WORD   EM29
  (6)  017302  007770                    .WORD   ERR29
 6845  017304                  65$:      ESCAPE  SEG
  (3)  017304  104410                    TRAP    C$ESCAPE
  (3)  017306  000010                    .WORD   10000$-.
 6846  017310  000241                    CLC                     ;CLEAR CARRY
 6847  017312  106105                    ROLB    R5              ;SHIFT BIT IN R5
 6848  017314  001343                    BNE     64$             ;IF R5=0 THEN DONE
 6849  017316                            ENDSEG
  (3)  017316                  10000$:
  (3)  017316  104405                    TRAP    C$ESEG
 6850  017320  012705  000001            MOV     #1,R5           ;START WITH BIT 0
 6851                        ;69$.       COM     R5              ;CHANGE TO FLOATING ZERO
 6852  017324                            BGNSEG
  (3)  017324  104404                    TRAP    C$BSEG
 6853  017326                  67$:
 6854  017326  005105                    COM     R5
 6855  017330  010561  000004            MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 6856  017334                            ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  017334  004537  003230            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6857  017340  122105                    122100!5                ;MOV DATA TO IBUS* REGISTER 5
 6858  017342                            ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  017342  004537  003230            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6859  017346  021125                    21005!<5*20>            ;READ FROM IBUS* REGISTER 5
 6860  017350  116104  000005            MOVB    5(R1),R4        ;PUT ''FOUND'' INTO R4
 6861  017354  120504                    CMPB    R5,R4           ;DATA CORRECT?
 6862  017356  001414                    BEQ     68$             ;BR IF YES
 6863  017360                            BEPROR  29              ;ERROR
  (5)  017400  104455                    TRAP    C$ERDF
  (6)  017402  000035                    .WORD   29
  (6)  017404  005571                    .WORD   EM29
  (6)  017406  007770                    .WORD   ERR29
 6864  017410                  68$:      ESCAPE  SEG
  (3)  017410  104410                    TRAP    C$ESCAPE
  (3)  017412  000012                    .WORD   10001$-.
 6865  017414  005105                    COM     R5              ;CHANGE TO FLOATING 1
 6866  017416  000241                    CLC                     ;CLEAR CARRY
 6867  017420  106105                    ROLB    R5              ;SHIFT BIT IN R5
 6868  017422  001341                    BNE     67$             ;IF R5=0 THEN DONE
 6869  017424                            ENDSEG
  (3)  017424                  10001$:
  (3)  017424  104405                    TRAP    C$ESEG
 6870  017426                  ENDTST
  (3)  017426                  L10102:
  (3)  017426  104401                    TRAP    C$ETST
 6871
 6872  017430                            BADHEAD
  (2)                                    ;******************** TEST 26 ********************
 6873                                    ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
 6874                                    ;*FLOAT A 1 THOUGH IBUS REGISTER 6
 6875                                    ;*FLOAT A 0 THROUGH IBUS REGISTER 6
 6876  017430                            BADHEAD
  (2)                                    ;******************** TEST 26 ********************
 6877
 6878  017430                  BGNTST
  (3)  017430                  T26::
```

```
6879  017430                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1) 017430  004537  003142                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
6880  017434  012737  000006  002624          MOV     #6,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
6881  017442  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
6882  017446                                  MYINT
  (1) 017446  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6883  017452                                  BGNSEG
  (3) 017452  104404                          TRAP    C$BSEG
6884  017454                            64$:
6885  017454  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
6886  017460                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1) 017460  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6887  017464  122106                          122100!6                ;MOV DATA TO IBUS* REGISTER 6
6888  017466                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1) 017466  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6889  017472  021145                          21005!<6*20>            ;READ FROM IBUS* REGISTER 6
6890  017474  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
6891  017500  120504                          CMPB    R5,R4           ;DATA CORRECT?
6892  017502  001414                          BEQ     65$             ;BR IF YES
6893  017504                                  BERROR  29              ;ERROR
  (5) 017524  104455                          TRAP    C$ERDF
  (6) 017526  000035                          .WORD   29
  (6) 017530  005571                          .WORD   EM29
  (6) 017532  007770                          .WORD   ERR29
6894  017534                            65$:  ESCAPE  SEG
  (3) 017534  104410                          TRAP    C$ESCAPE
  (3) 017536  000010                          .WORD   10000$-.
6895  017540  000241                          CLC                     ;CLEAR CARRY
6896  017542  106105                          ROLB    R5              ;SHIFT BIT IN R5
6897  017544  001343                          BNE     64$             ;IF R5=0 THEN DONE
6898  017546                                  ENDSEG
  (3) 017546                            10000$:
  (3) 017546  104405                          TRAP    C$ESEG
6899  017550  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
6900                                    ;69$: COM     R5              ;CHANGE TO FLOATING ZERO
6901  017554                                  BGNSEG
  (3) 017554  104404                          TRAP    C$BSEG
6902  017556                            67$:
6903  017556  005105                          COM     R5
6904  017560  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
6905  017564                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1) 017564  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6906  017570  122106                          122100!6                ;MOV DATA TO IBUS* REGISTER 6
6907  017572                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1) 017572  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6908  017576  021145                          21005!<6*20>            ;READ FROM IBUS* REGISTER 6
6909  017600  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
6910  017604  120504                          CMPB    R5,R4           ;DATA CORRECT?
6911  017606  001414                          BEQ     68$             ;BR IF YES
6912  017610                                  BERROR  29              ;ERROR
  (5) 017630  104455                          TRAP    C$ERDF
  (6) 017632  000035                          .WORD   29
  (6) 017634  005571                          .WORD   EM29
  (6) 017636  007770                          .WORD   ERR29
6913  017640                            68$:  ESCAPE  SEG
  (3) 017640  104410                          TRAP    C$ESCAPE
```

```
  (3)  017642  000012                        .WORD   10001$-.
6914   017644  005105                        COM     R5              ;CHANGE TO FLOATING 1
6915   017646  000241                        CLC                     ;CLEAR CARRY
6916   017650  106105                        ROLB    R5              ;SHIFT BIT IN R5
6917   017652  001341                        BNE     67$             ;IF R5=0 THEN DONE
6918   017654                                ENDSEG
  (3)  017654                          10001$:
  (3)  017654  104405                        TRAP    C$ESEG
6919   017656                        ENDTST
  (3)  017656                        L10103:
  (3)  017656  104401                        TRAP    C$ETST
6920
6921   017660                                BADHEAD
  (2)                                         ;******************** TEST 27 ********************
6922                                          ;*MICRO PROCEOR IBUS* REGISTER WRITE/READ TEST
6923                                          ;*FLOAT A 1 THOUGH IBUS* REGISTER 7
6924                                          ;*FLOAT A 0 THROUGH IBUS* REGISTER 7
6925   017660                                BADHEAD
  (2)                                         ;******************** TEST 27 ********************
6926
6927   017660                        BGNTST
  (3)  017660                        T27::
6928   017660                                MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  017660  004537  003142                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
6929   017664  012737  000007  002624        MOV     #7,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
6930   017672  012705  000001                MOV     #1,R5           ;START WITH BIT 0
6931   017676                                MYINT
  (1)  017676  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6932   017702                                BGNSEG
  (3)  017702  104404                        TRAP    C$BSEG
6933   017704                          64$:
6934   017704  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
6935   017710                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  017710  004537  003230                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6936   017714  122107                        122100!7                ;MOV DATA TO IBUS* REGISTER 7
6937   017716                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  017716  004537  003230                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6938   017722  021165                        21005!<7*20>            ;READ FROM IBUS* REGISTER 7
6939   017724  116104  000005                MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
6940   017730  120504                        CMPB    R5,R4           ;DATA CORRECT?
6941   017732  001414                        BEQ     65$             ;BR IF YES
6942   017734                                BERROR  29              ;ERROR
  (5)  017754  104455                        TRAP    C$ERDF
  (6)  017756  000035                        .WORD   29
  (6)  017760  005571                        .WORD   EM29
  (6)  017762  007770                        .WORD   ERR29
6943   017764                          65$:  ESCAPE  SEG
  (3)  017764  104410                        TRAP    C$ESCAPE
  (3)  017766  000010                        .WORD   10000$-.
6944   017770  000241                        CLC                     ;CLEAR CARRY
6945   017772  106105                        ROLB    R5              ;SHIFT BIT IN R5
6946   017774  001343                        BNE     64$             ;IF R5=0 THEN DONE
6947   017776                                ENDSEG
  (3)  017776                          10000$:
  (3)  017776  104405                        TRAP    C$ESEG
6948   020000  012705  000001                MOV     #1,R5           ;START WITH BIT 0
```

```
 6949
 6950  020004
  (3)  020004  104404
 6951  020006
 6952  020006  005105
 6953  020010  010561  000004
 6954  020014
  (1)  020014  004537  003230
 6955  020020  122107
 6956  020022
  (1)  020022  004537  003230
 6957  020026  021165
 6958  020030  116104  000005
 6959  020034  120504
 6960  020036  001414
 6961  020040
  (5)  020060  104455
  (6)  020062  000035
  (6)  020064  005571
  (6)  020066  007770
 6962  020070
  (3)  020070  104410
  (3)  020072  000012
 6963  020074  005105
 6964  020076  000241
 6965  020100  106105
 6966  020102  001341
 6967  020104
  (3)  020104
  (3)  020104  104405
 6968  020106
  (3)  020106
  (3)  020106  104401
 6969
 6970  020110
  (2)
 6971
 6972
 6973
 6974  020110
  (2)
 6975
 6976  020110
  (3)  020110
 6977  020110
  (1)  020110  004537  003142
 6978  020114  012705  000001
 6979  020120  005002
 6980  020122
  (1)  020122  013701  002716
 6981  020126
  (3)  020126  104404
 6982  020130  010203
 6983  020132  010561  000004
 6984  020136  042737  000017  020154
 6985  020144  050337  020154

;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
        BGNSEG
        TRAP    C$BSEG
67$:
        COM     R5
        MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
        JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
        122100!7                ;MOV DATA TO IBUS* REGISTER 7
        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
        JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
        21005!<7*20>            ;READ FROM IBUS* REGISTER 7
        MOVB    5(R1),R4        ;PUT ''FOUND'' INTO R4
        CMPB    R5,R4           ;DATA CORRECT?
        BEQ     68$             ;BR IF YES
        BERROR  29              ;ERROR
        TRAP    C$FRDF
        .WORD   29
        .WORD   EM29
        .WORD   ERR29
68$:    ESCAPE  SEG
        TRAP    C$ESCAPE
        .WORD   10001$-.
        COM     R5              ;CHANGE TO FLOATING 1
        CLC                     ;CLEAR CARRY
        ROLB    R5              ;SHIFT BIT IN R5
        BNE     67$             ;IF R5=0 THEN DONE
        ENDSEG
10001$:
        TRAP    C$ESEG
ENDTST
L10104:
        TRAP    C$ETST

        BADHEAD
        ;******************** TEST 28 ********************
        ;*MICRO PROCESSOR IBUS DUAL ADDRESS TEST
        ;*WRITE ALL IBUS REGISTERS WITH INCREMENTING PATTERN
        ;*READ ALL IBUS REGISTERS TO VERIFY CORRECT ADDRESSING
        BADHEAD
        ;******************** TEST 28 ********************

BGNTST
T28::
        MSTCLR                  ;MASTER CLEAR M8200,4,7
        JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
        MOV     #1,R5           ;START WITH A ONE
        CLR     R2              ;R2 CONTAINS ADDRESS OF REGISTER
        MYINT
        MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
        BGNSEG
        TRAP    C$BSEG
1$:     MOV     R2,R3           ;R3=REGISTER ADDRESS
        MOV     R5,4(R1)        ;WRITE DATA TO PORT4
        BIC     #17,5$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
        BIS     R3,5$           ;ADD ADDRESS TO INSTRUCTION
```

```
     020150                               ROMCLK                 ;NEXT WORD IS INSTRUCTION,  BBN
 (1) 020150  004537  003230               JSR      R5,.ROMCLK    ;CLOCK INSTRUCTION
     020154  122100               5$:     122100                 ;MOVE DATA TO IBUS REGISTER
     020156  006303                       ASL      R3            ;SHIFT ADDRESS
     020160  006303                       ASL      R3            ;4 TIMES TO GET
     020162  006303                       ASL      R3            ;IT TO BITS 4-7
     020164  006303                       ASL      R3            ;OF NEXT INSTRUCTION
     020166  042737  000360  020204       BIC      #360,6$       ;CLEAR ADDRESS FIELD
     020174  050337  020204               BIS      R3,6$         ;ADD ADDRESS TO INSTRUCTION
     020200                               ROMCLK                 ;NEXT WORD IS INSTRUCTION,  BBN
 (1) 020200  004537  003230               JSR      R5,.ROMCLK    ;CLOCK INSTRUCTION
     020204  021005               6$:     21005                  ;READ FROM IBUS REGISTER
     020206  116104  000005               MOVB     5(R1),R4      ;PUT 'FOUND'' IN R4
     020212  120504                       CMPB     R5,R4         ;IS DATA CORRECT?
     020214  001414                       BEQ      2$            ;BR IF YES
     020216                               BERROR   29            ;DATA ERROR
 (5) 020236  104455                       TRAP     C$ERDF
 (6) 020240  000035                       .WORD    29
 (6) 020242  005571                       .WORD    EM29
 (6) 020244  007770                       .WORD    ERR29
     020246                       2$:     ESCAPE   SEG
 (3) 020246  104410                       TRAP     C$ESCAPE
 (3) 020250  000014                       .WORD    10000$-.
     020252  005205                       INC      R5            ;INCREMENT PATTERN
     020254  005202                       INC      R2            ;INCREMENT REGISTER ADDRESS
     020256  022702  000010               CMP      #7+1,R2       ;LAST ADDRESS DONE?
     020262  001322                       BNE      1$            ;BR IF NO
     020264                               ENDSEG
 (3) 020264                       10000$:
 (3) 020264  104405                       TRAP     C$ESEG
     020266  012705  000001               MOV      #1,R5         ;RESTART PATTERN TO 1
     020272  005002                       CLR      R2            ;RESTART AT ADDRESS 0
     020274                               BGNSEG
 (3) 020274  104404                       TRAP     C$BSEG
     020276  005003                       CLR      R3            ;RESTART AT ADDRESS 0
     020300  042737  000360  020316  3$:  BIC      #360,7$       ;CLEAR ADDRESS FIELD OF INSTRUCTION
     020306  050337  020316               BIS      R3,7$         ;ADD ADDRESS TO INSTRUCTION
     020312                               ROMCLK                 ;NEXT WORD IS INSTRUCTION,  BBN
 (1) 020312  004537  003230               JSR      R5,.ROMCLK    ;CLOCK INSTRUCTION
     020316  021005               7$:     21005                  ;READ FROM IBUS REGISTER
     020320  116104  000005               MOVB     5(R1),R4      ;PUT 'FOUND'' IN $GDDAT
     020324  120504                       CMPB     R5,R4         ;DATA CORRECT?
     020326  001414                       BEQ      4$            ;BR IF YES
     020330                               BERROR   30            ;DUAL ADDRESSING ERROR
 (5) 020350  104455                       TRAP     C$ERDF
 (6) 020352  000036                       .WORD    30
 (6) 020354  004337                       .WORD    EM30
 (6) 020356  010052                       .WORD    ERR30
     020360                       4$:     ESCAPE   SEG
 (3) 020360  104410                       TRAP     C$ESCAPE
 (3) 020362  000020                       .WORD    10001$-.
     020364  005205                       INC      R5            ;INCREMENT PATTERN
     020366  005202                       INC      R2            ;NEXT ADDRESS
     020370  062703  000020               ADD      #20,R3        ;ADD 1 TO ADDRESS IN R3(SHIFTED 4 TIMES)
     020374  022702  000010               CMP      #7+1,R2       ;LAST ADDRESS DONE?
     020400  001337                       BNE      3$            ;BR IF NO
```

```
7024  020402                          ENDSEG
 (3)  020402                  10001$:
 (3)  020402  104405                  TRAP    C$ESEG
7025  020404                  ENDTST
 (3)  020404                  L10105:
 (3)  020404  104401                  TRAP    C$ETST
7026
7027  020406                          BADHEAD
 (2)                                  ;******************** TEST 29 ********************
7028                                  ;*MICRO PROCESSOR BR REGISTER TEST
7029                                  ;*FLOAT A 1 THOUGH THE BR
7030                                  ;*FLOAT A 0 THOUGH THE BR
7031  020406                          BADHEAD
 (2)                                  ;******************** TEST 29 ********************
7032
7033  020406                  BGNTST
 (3)  020406                  T29::
7034                                                          ;R1 CONTAINS BASE M8200,4,7 ADDRESS
7035  020406                          MSTCLR                  ;MASTER CLEAR COMM. MICRO-PROCCESSOR FAMILY
 (1)  020406  004537  003142          JSR     R5,.MSTCLR               ;CLEAR M8200,4,7
7036  020412  012702  000001          MOV     #1,R2           ;START PATTERN WITH BIT0
7037  020416                          MYINT
 (1)  020416  013701  002716          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7038  020422                          BGNSEG
 (3)  020422  104404                  TRAP    C$BSEG
7039  020424                  64$:
7040  020424  010261  000004          MOV     R2,4(R1)        ;WRITE PATTERN IN PORT4
7041  020430                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  020430  004537  003230          JSR     R5,.ROMCLK               ;CLOCK INSTRUCTION
7042  020434  120500                  120500                  ;MOVE DATA TO THE BR REGISTER
7043  020436                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  020436  004537  003230          JSR     R5,.ROMCLK               ;CLOCK INSTRUCTION
7044  020442  061225                  061225                  ;MOVE BR TO PORT 5
7045  020444  116104  000005          MOVB    5(R1),R4        ;PUT ''FOUND'' IN R4
7046  020450  120204                  CMPB    R2,R4           ;IS DATA CORRECT?
7047  020452  001414                  BEQ     65$             ;BR IF YES
7048  020454                          BERROR  3               ;DATA ERROR
 (5)  020474  104455                  TRAP    C$ERDF
 (6)  020476  000003                  .WORD   3
 (6)  020500  004402                  .WORD   EM3
 (6)  020502  006174                  .WORD   ERR3
7049  020504                  65$:    ESCAPE  SEG
 (3)  020504  104410                  TRAP    C$ESCAPE
 (3)  020506  000010                  .WORD   10000$-.
7050  020510  000241                  CLC                     ;CLEAR CARRY
7051  020512  106102                  ROLB    R2              ;SHIFT BIT IN R2
7052  020514  001343                  BNE     64$             ;DONE IF R2=0
7053  020516                          ENDSEG
 (3)  020516                  10000$:
 (3)  020516  104405                  TRAP    C$ESEG
7054  020520  012702  000001          MOV     #1,R2           ;START PATTERN WITH BIT0
7055  020524                  69$:
7056  020524                          BGNSEG
 (3)  020524  104404                  TRAP    C$BSEG
7057  020526                  67$:
7058  020526  005102                  COM     R2
```

```
7059  020530  010261  000004                  MOV     R2,4(R1)        ;WRITE PATTERN IN PORT4
7060  020534                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  020534  004537  003230                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7061  020540  120500                          120500                  ;MOVE DATA TO THE BR REGISTER
7062  020542                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  020542  004537  003230                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7063  020546  061225                          061225                  ;MOVE BR TO PORT 5
7064  020550  116104  000005                  MOVB    5(R1),R4        ;PUT ''FOUND'' IN $GDDAT
7065  020554  010205                          MOV     R2,R5
7066  020556  120204                          CMPB    R2,R4           ;DATA CORRECT?
7067  020560  001414                          BEQ     68$             ;BR IF YES
7068  020562                                   BERROR  3               ;DATA ERROR
 (5)  020602  104455                          TRAP    C$ERDF
 (6)  020604  000003                          .WORD   3
 (6)  020606  004402                          .WORD   EM3
 (6)  020610  006174                          .WORD   ERR3
7069  020612                          68$:     ESCAPE  SEG
 (3)  020612  104410                          TRAP    C$ESCAPE
 (3)  020614  000106                          .WORD   10001$-.
7070  020616  052711  040000                  BIS     #40000,(R1)     ;SET MASTER CLEAR
7071  020622  105761  000001                  TSTB    1(R1)
7072  020626  001427                          BEQ     70$
7073  020630  100426                          BMI     70$
7074  020632  042711  040000                  BIC     #40000,(R1)
7075
7076                                           ;TO RUN THIS SECTION OF CODE YOU MUST TURN SW7 OF SWITCH PACK #E28
7077                                           ;OFF SO THAT M8207 NOT SELFSTARTING.
7078
7079  020636                                   ROMCLK                  ;PUT BR IN PORT5
 (1)  020636  004537  003230                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7080  020642  061225                          061225
7081  020644  116104  000005                  MOVB    5(R1),R4        ;READ IT
7082  020650  001416                          BEQ     70$             ;IF ZERO, OK
7083  020652  005005                          CLR     R5
7084  020654                                   BERROR  3               ;MASTER CLEAR
 (5)  020674  104455                          TRAP    C$ERDF
 (6)  020676  000003                          .WORD   3
 (6)  020700  004402                          .WORD   EM3
 (6)  020702  006174                          .WORD   ERR3
7085  020704                                   CKLOOP
 (3)  020704  104406                          TRAP    C$CLP1
7086                                                                   ;FAILED TO CLEAR
7087  020706  105061  000001          70$:     CLRB    1(R1)           ;BRG
7088  020712  005102                          COM     R2              ;CHANGE BACK TO A ONE
7089  020714  000241                          CLC                     ;CLEAR CARRY
7090  020716  106102                          ROLB    R2              ;SHIFT BIT IN R5
7091  020720  001302                          BNE     67$             ;DONE IF R5=0
7092  020722                                   ENDSEG
 (3)  020722                          10001$:
 (3)  020722  104405                          TRAP    C$ESEG
7093  020724                          ENDTST
 (3)  020724                          L10106:
 (3)  020724  104404                          TRAP    C$ETST
7094
7095  020726                                   BADHEAD
 (2)                                         ;****************** TEST 30 ******************
```

```
7096
7097                                    ;*SCRATCH PAD TEST
7098                                    ;*FLOAT A 1 THOUGH EACH SCRATCH PAD LOCATION
7099  020726                            ;*FLOAT A 0 THOUGH EACH SCRATCH PAD LOCATION
                                        BADHEAD
 (2)                                    ;******************** TEST 30 ********************
7100
7101  020726                    BGNTST
 (3)  020726                    T30::
7102  020726                            MYINT
 (1)  020726  013701  002716            MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7103  020732                            MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  020732  004537  003142            JSR     R5,..MSTCLR     ;CLEAR M8200,4,7
7104  020736  005002                    CLR     R2              ;START AT ADDRESS ZERO
7105  020740  012705  000001            MOV     #1,R5           ;START WITH BIT0
7106  020744                            BGNSUB
 (3)  020744                    T30.1:
 (3)  020744  104402                    TRAP    C$BSUB
7107  020746                    1$:     BGNSEG
 (3)  020746  104404                    TRAP    C$BSEG
7108  020750  042737  000017  020772  64$:    BIC     #17,65$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
7109  020756  050237  020772            BIS     R2,65$          ;ADD ADDRESS TO INSTRUCTION
7110  020762  010561  000004            MOV     R5,4(R1)        ;WRITE PATTERN IN PORT4
7111  020766                            ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  020766  004537  003230            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7112  020772  123100                    65$:    123100          ;WRITE SCRATCH PAD(ADDRESS IN R2)
7113  020774  042737  000017  021012    BIC     #17,66$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
7114  021002  050237  021012            BIS     R2,66$          ;ADD ADDRESS TO INSTRUCTION
7115  021006                            ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  021006  004537  003230            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7116  021012  040600                    66$:    040600          ;MOVE SP TO BR
7117  021014                            ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  021014  004537  003230            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7118  021020  061225                    061225                  ;MOVE BR TO PORT5
7119  021022  010537  002636            MOV     R5,$GDDAT               ;PUT 'EXPECTED'' IN $GDDAT
7120  021026  116104  000005            MOVB    5(R1),R4        ;PUT 'FOUND'' IN R4
7121  021032  123704  002636            CMPB    $GDDAT,R4               ;DATA CORRECT
7122  021036  001414                    BEQ     67$             ;BR IF YES
7123  021040                            RERROR  4               ;DATA ERROR
 (5)  021060  104455                    TRAP    C$ERDF
 (6)  021062  000004                    .WORD   4
 (6)  021064  004430                    .WORD   EM4
 (6)  021066  006252                    .WORD   ERR4
7124  021070                    67$:    ESCAPE  SEG
 (3)  021070  104410                    TRAP    C$ESCAPE
 (3)  021072  000010                    .WORD   10000$-.
7125  021074  000241                    CLC                     ;CLEAR CARRY
7126  021076  106105                    ROLB    R5              ;SHIFT BIT IN R5
7127  021100  001323                    BNE     64$             ;DONE IF R5=0
7128  021102                            ENDSEG
 (3)  021102                    10000$:
 (3)  021102  104405                    TRAP    C$ESEG
7129  021104  012705  000001            MOV     #1,R5           ;START WITH BIT0
7130  021110                            BGNSEG
 (3)  021110  104404                    TRAP    C$BSEG
7131
7132  021112  005105            73$:    COM     R5              ;CHANGE TO FLOATING ZERO
```

```
 7133  021114  042737  000017  021136  69$:    BIC     #17,70$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7134  021122  050237  021136                   BIS     R2,70$           ;ADD ADDRESS TO INSTRUCTION
 7135  021126  010561  000004                   MOV     R5,4(R1)         ;WRITE PATTERN IN PORT4
 7136  021132                                   ROMCLK                   ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  021132  004537  003230                   JSR     R5,..ROMCLK              ;CLOCK INSTRUCTION
 7137  021136  123100                   70$:    123100                   ;WRITE SCRATCH PAD(ADDRESS IN R2)
 7138  021140  042737  000017  021156           BIC     #17,71$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7139  021146  050237  021156                   BIS     R2,71$           ;ADD ADDRESS TO INSTRUCTION
 7140  021152                                   ROMCLK                   ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  021152  004537  003230                   JSR     R5,..ROMCLK              ;CLOCK INSTRUCTION
 7141  021156  040600                   71$:    040600                   ;MOVE SP TO BR
 7142  021160                                   ROMCLK                   ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  021160  004537  003230                   JSR     R5,..ROMCLK              ;CLOCK INSTRUCTION
 7143  021164  061225                           061225                   ;MOVE BR TO PORT5
 7144  021166  010537  002636                   MOV     R5,$GDDAT                ;PUT "EXPECTED" IN $GDDAT
 7145  021172  116104  00000?                   MOVB    5(R1),R4         ;PUT "FOUND" IN $GDDAT
 7146  021176  123704  002636                   CMPB    $GDDAT,R4                ;DATA CORRECT?
 7147  021202  001414                           BEQ     72$              ;BR IF YES
 7148  021204                                   RERROR  4                ;DATA ERROR
  (5)  021224  104455                           TRAP    C$ERDF
  (6)  021226  000004                           .WORD   4
  (6)  021230  004430                           .WORD   EM4
  (6)  021232  006252                           .WORD   ERR4
 7149  021234                           72$:    ESCAPE  TST
  (3)  021234  104410                           TRAP    C$ESCAPE
  (3)  021236  000032                           .WORD   L10107-.
 7150  021240  005105                           COM     R5               ;CHANGE BACK TO A ONE
 7151  021242  000241                           CLC                      ;CLEAR CARRY
 7152  021244  106105                           ROLB    R5               ;SHIFT BIT IN R5
 7153  021246  001321                           BNE     73$              ;DONE IF R5=0
 7154  021250                           ENDSEG
  (3)  021250                           10001$:
  (3)  021250  104405                           TRAP    C$ESEG
 7155  021252  012705  000001                   MOV     #1,R5            ;RESTART AT BIT 0
 7156  021256  005202                           INC     R2               ;NEXT SP ADDRESS
 7157  021260  022702  000020                   CMP     #20,R2           ;LAST ADDRESS?
 7158  021264  001230                           BNE     1$               ;BR IF NO
 7159  021266                                   ENDSUB
  (3)  021266                           _10110:
  (3)  021266  104403                           TRAP    C$ESUB
 7160  021270                           ENDTST
  (3)  021270                           L10107:
  (3)  021270  104401                           TRAP    C$ETST
 7161
 7162  021272                                   BADHEAD
  (2)                                           ;******************** TEST 31 ********************
 7163                                           ;*SCRATCH PAD DUAL ADDRESSING TEST
 7164                                           ;*WRITE AN INCREMENTING PATTERN IN ALL SP LOCATIONS
 7165                                           ;*READ ALL SP LOCATIONS TO VERIFY CORRECT ADDRESSING
 7166  021272                                   BADHEAD
  (2)                                           ;******************** TEST 31 ********************
 7167
 7168  021272                           BGNTST
  (3)  021272                           T31::
 7169  021272                                   MSTCLR                   ;MASTER CLEAR M8200,4,7
  (1)  021272  004537  003142                   JSR     R5,..MSTCLR              ;CLEAR M8200,4,7
```

```
 7170  021276  012705  000001            MOV     #1,R5           ;START WITH A 1
 7171  021302  005003                    CLR     R3              ;ADDRESS 0
 7172  021304                            MYINT
   (1) 021304  013701  002716            MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7173  021310                            BGNSEG
   (3) 021310  104404                    TRAP    C$BSEG
 7174  021312  010302              1$:   MOV     R3,R2           ;MOVE ADDRESS TO R2
 7175  021314  042737  000017  021336    BIC     #17,2$          ;CLEAR ADDRESS FIELD
 7176  021322  050237  021336            BIS     R2,2$           ;ADD ADDRESS TO INSTRUCTION
 7177  021326  010561  000004            MOV     R5,4(R1)        ;WRITE PATTERN IN PORT4
 7178  021332                            ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
   (1) 021332  004537  003230            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7179  021336  123100              2$:   123100                  ;WRITE SP(ADDRESS IN R2)
 7180  021340  042737  000017  021356    BIC     #17,3$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
```

```
 7182  021346  050237  021356                BIS     R2,3$           ;ADD ADDRESS TO INSTRUCTION
 7183  021352                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  021352  004537  003230                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
```

```
7185  021356  060600                    3$:     60600                   ;MOVE SP TO BR
7186  021360                            ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  021360  004537  003230            JSR     R5,..ROMCLK                     ;CLOCK INSTRUCTION
7187  021364  061225                    61225                           ;MOVE BR TO PORT5
7188  021366  010537  002636            MOV     R5,$GDDAT                       ;PUT ''EXPECTED'' IN $GDDAT
7189  021372  116104  000005            MOVB    5(R1),R4                ;PUT ''FOUND'' IN R4
7190  021376  123704  002636            CMPB    $GDDAT,R4                       ;DATA CORRECT
7191  021402  001414                    BEQ     4$                      ;BR IF YES
7192  021404                            RERROR  4                       ;DATA ERROR
 (5)  021424  104455                    TRAP    C$ERDF
 (6)  021426  000004                    .WORD   4
 (6)  021430  004430                    .WORD   EM4
 (6)  021432  006252                    .WORD   ERR4
7193  021434                            4$:     ESCAPE  SEG
 (3)  021434  104410                    TRAP    C$ESCAPE
 (3)  021436  000014                    .WORD   10000$-.
7194  021440  005205                    INC     R5                      ;INCREMENT PATTERN
7195  021442  005203                    INC     R3                      ;NEXT ADDRESS
7196  021444  022703  000020            CMP     #20,R3                  ;LAST ADDRESS DONE?
7197  021450  001320                    BNE     1$                      ;BR IF NO
7198  021452                            ENDSEG
 (3)  021452                            10000$:
 (3)  021452  104405                    TRAP    C$ESEG
7199  021454  012705  000001            MOV     #1,R5                   ;RESTART PATTERN AT 1
7200  021460  005003                    CLR     R3                      ;RESTART AT ADDRESS ZERO
7201  021462                            BGNSEG
 (3)  021462  104404                    TRAP    C$BSEG
7202  021464  010302                    5$:     MOV     R3,R2           ;PUT ADDRESS IN R2
7203  021466  042737  000017  021504    69$:    BIC     #17,6$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7204  021474  050237  021504            BIS     R2,6$                   ;ADD ADDRESS TO INSTRUCTION
7205  021500                            ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  021500  004537  003230            JSR     R5,..ROMCLK                     ;CLOCK INSTRUCTION
7206  021504  060600                    6$:     60600                   ;MOV SP TO BR
7207  021506                            ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  021506  004537  003230            JSR     R5,..ROMCLK                     ;CLOCK INSTRUCTION
7208  021512  061225                    61225                           ;MOV BR TO PORT5
7209  021514  010537  002636            MOV     R5,$GDDAT                       ;PUT ''EXPECTED'' IN $GDDAT
7210  021520  116104  000005            MOVB    5(R1),R4                ;PUT ''FOUND'' IN $GDDAT
7211  021524  123704  002636            CMPB    $GDDAT,R4                       ;DATA CORRECT?
7212  021530  001414                    BEQ     7$                      ;BR IF YES
7213  021532                            RERROR  5                       ;SP ADDRESSING ERROR
 (5)  021552  104455                    TRAP    C$ERDF
 (6)  021554  000005                    .WORD   5
 (6)  021556  004456                    .WORD   EM5
 (6)  021560  006334                    .WORD   ERR5
7214  021562                            7$:     ESCAPE  SEG
 (3)  021562  104410                    TRAP    C$ESCAPE
 (3)  021564  000014                    .WORD   10001$-.
7215  021566  005205                    INC     R5                      ;INCREMENT PATTERN
7216  021570  005203                    INC     R3                      ;NEXT ADDRESS
7217  021572  022703  000020            CMP     #20,R3                  ;LAST ADDRESS DONE?
7218  021576  001332                    BNE     5$                      ;BR IF NO
7219  021600                            ENDSEG
 (3)  021600                            10001$:
 (3)  021600  104405                    TRAP    C$ESEG
7220  021602                            ENDTST
```

```
  (3)  021602                          L10111:
  (3)  021602  104401                          TRAP    C$ETST
7221
7222  021604                                   BADHEAD
  (2)                                          ;********************* TEST 32 *********************
7223                                           ;*INTERRUPT TEST
7224                                           ;*TEST THAT DEVICE CAN INTERRUPT TO VECTOR A
7225  021604                                   BADHEAD
  (2)                                          ;********************* TEST 32 *********************
7226
7227  021604                          BGNTST
  (3)  021604                          T32::
7228  021604                                   MYINT
  (1)  021604  013701  002716                 MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7229  021610                                   BRESET                  ;BUS RESET
  (3)  021610  104433                         TRAP    C$RESET
7230  021612  005011                         CLR     (R1)            ;CLEAR RUN
7231  021614  004537  003536                 JSR     R5,SETVEC       ;SET UP VECTORS
7232  021620  021740                         3$                      ;XX0
7233  021622  021712                         2$                      ;XX4
7234  021624  000340  000340                 .WORD   340,340         ;LEVEL 7
7235  021630                          1$:     SETPRI  #PRI07          ;PS = LEVEL 7
  (3)  021630  012700  000340                 MOV     #PRI07,R0
  (3)  021634  104441                         TRAP    C$SPRI
7236  021636  012761  000200  000004         MOV     #200,4(R1)      ;WRITE PORT4
7237  021644                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  021644  004537  003230                 JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7238  021650  121111                         121111                  ;SET BR RQ IN IBUS* REG 11
7239  021652                                   SETPRI  #PRI00          ;ALLOW INTERRUPT
  (3)  021652  012700  000000                 MOV     #PRI00,R0
  (3)  021656  104441                         TRAP    C$SPRI
7240  021660  000240                         NOP
7241  021662                                   ERROR   31              ;NO INTERRUPT
  (5)  021700  104455                         TRAP    C$ERDF
  (6)  021702  000037                         .WORD   31
  (6)  021704  005276                         .WORD   EM31
  (6)  021706  010130                         .WORD   ERR31
7242  021710  000415                         BR      4$
7243  021712                          2$:     ERROR   32              ;WRONG VECTOR
  (5)  021730  104455                         TRAP    C$ERDF
  (6)  021732  000040                         .WORD   32
  (6)  021734  005325                         .WORD   EM32
  (6)  021736  010156                         .WORD   ERR32
7244  021740  062706  000004          3$:     ADD     #4,SP   ;RESET STACK
7245  021744                          4$:
7246  021744                          ENDTST
  (3)  021744                          L10112:
  (3)  021744  104401                          TRAP    C$ETST
7247
7248  021746                                   BADHEAD
  (2)                                          ;********************* TEST 33 *********************
7249                                           ;*INTERRUPT TEST
7250                                           ;*TFST THAT DEVICE CAN INTERRUPT TO VECTOR B
7251  021746                                   BADHEAD
  (2)                                          ;********************* TEST 33 *********************
7252
```

```
7253  021746                          BGNTST
 (3)  021746                          T33::
7254  021746                                MYINT
 (1)  021746  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7255  021752                                MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  021752  004537  003142                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
7256  021756  004537  003536                JSR     R5,SETVEC       ;SET UP VECTORS
7257  021762  022054                        2$                      ;XX0
7258  021764  022102                        3$                      ;XX4
7259  021766  000340  000340                .WORD   340,340         ;LEVEL 7
7260  021772                          1$:   SETPRI  #PRI07          ;PS = LEVEL 7
 (3)  021772  012700  000340                MOV     #PRI07,R0
 (3)  021776  104441                        TRAP    C$SPRI
7261  022000  012761  000300  000004        MOV     #300,4(R1)      ;WRITE PORT4
7262  022006                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  022006  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7263  022012  121111                        121111                  ;SET BR RQ IN IBUS* REG 11
7264  022014                                SETPRI  #PRI00          ;ALLOW INTERRUPT
 (3)  022014  012700  000000                MOV     #PRI00,R0
 (3)  022020  104441                        TRAP    C$SPRI
7265  022022  000240                        NOP
7266  022024                                ERROR   31              ;NO INTERRUPT
 (5)  022042  104455                        TRAP    C$ERDF
 (6)  022044  000037                        .WORD   31
 (6)  022046  005276                        .WORD   EM31
 (6)  022050  010130                        .WORD   ERR31
7267  022052  000415                        BR      4$
7268  022054                          2$:   ERROR   32              ;WRONG VECTOR
 (5)  022072  104455                        TRAP    C$ERDF
 (6)  022074  000040                        .WORD   32
 (6)  022076  005325                        .WORD   EM32
 (6)  022100  010156                        .WORD   ERR32
7269  022102  062706  000004          3$:   ADD     #4,SP   ;RESET STACK
7270  022106                          4$:
7271  022106                          ENDTST
 (3)  022106                          L10113:
 (3)  022106  104401                        TRAP    C$ETST
7272
7273  022110                                BADHEAD
 (2)                                        ;******************** TEST 34 ********************
7274                                        ;*PRIORITY INTERRUPT TEST
7275                                        ;*SET PS TO ALL BR LEVELS EQUAL OR GREATER THAN
7276                                        ;*THE M8200,4,7 LEVEL. VERIFY THAT M8200,4,7 DOES NOT INTERRUPT
7277  022110                                BADHEAD
 (2)                                        ;******************** TEST 34 ********************
7278
7279  022110                          BGNTST
 (3)  022110                          T34::
7280  022110                                MYINT
 (1)  022110  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7281  022114                                MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  022114  004537  003142                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
7282  022120  012704  000340                MOV     #340,R4         ;PUT LEVEL 7 IN R2
7283  022124                                SETPRI  R4              ;SET PRIORITY TO 7
 (3)  022124  010400                        MOV     R4,R0
 (3)  022126  104441                        TRAP    C$SPRI
```

```
 7284  022130  013705  002700                MOV     STAT1,R5        ;GET BR LEVEL OF M8200,4,7
 7285  022134  006205                        ASR     R5              ;SHIFT R5 4 TIMES
 7286  022136  006205                        ASR     R5              ;TO GET PROPER LEVEl
 7287  022140  006205                        ASR     R5
 7288  022142  006205                        ASR     R5
 7289  022144  042705  177437                BIC     #177437,R5      ;CLEAR UNWANTED BITS
 7290  022150  010537  002636                MOV     R5,$GDDAT
 7291  022154  004537  003536                JSR     R5,SETVEC       ;SET UP VECTORS
 7292  022160  022224                        2$                      ;A VECTOR
 7293  022162  022224                        2$                      ;B VECTOR
 7294  022164  000340  000340                .WORD   340,340         ;PRIORITY 7
 7295  022170  012761  C00200  000004  4$:   MOV     #200,4(R1)      ;LOAD PORT4
 7296  022176                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  022176  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7297  022202  121111                        121111                  ;SET BR REQUEST
 7298  022204                        5$:     SETPRI  R4              ;PUT LEVEL IN R2 IN PS
  (3)  022204  010400                        MOV     R4,R0
  (3)  022206  104441                        TRAP    C$SPRI
 7299  022210  000240                        NOP
 7300  022212  020504                        CMP     R5,R4           ;IS PRESENT PS LEVEL = TO M8200,4,7 LEVEL
 7301  022214  001420                        BEQ     1$              ;BR IF YES
 7302  022216  162704  000040                SUB     #40,R4          ;NO GET NEXT LOWER LEVEL IN R2
 7303  022222  000770                        BR      5$              ;AND CONTINUE WITH TEST
 7304  022224                        2$:     BRESET
  (3)  022224  104433                        TRAP    C$RESET
 7305  022226                                ERROR   33              ;ERROR UNEXPECTED INTERRUPT
  (5)  022244  104455                        TRAP    C$ERDF
  (6)  022246  000041                        .WORD   33
  (6)  022250  005364                        .WORD   EM33
  (6)  022252  010204                        .WORD   ERR33
 7306  022254  000002                        RTI
 7307  022256                        1$:     MSTCLR
  (1)  022256  004537  003142                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 7308  022262                                ENDTST
  (3)  022262                        L10114:
  (3)  022262  104401                .       TRAP    C$ETST
 7309
 7310  022264                                BADHEAD
  (2)                                       ;******************** TEST 35 ********************
 7311                                       ;*PRIORITY INTERRUPT TESTS
 7312                                       ;*SET PS TO ALL BR LEVELS LESS THAN THE M8200,4,7 LEVEL
 7313                                       ;*VERIFY THAT M8200,4,7 WILL INTERRUPT
 7314  022264                                BADHEAD
  (2)                                       ;******************** TEST 35 ********************
 7315
 7316  022264                        BGNTST
  (3)  022264                        T35::
 7317  022264                                MYINT
  (1)  022264  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7318  022270                                MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  022270  004537  003142                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 7319  022274  012704  000340                MOV     #340,R4         ;PUT LEVEL 7 IN R2
 7320  022300                                SETPRI  R4              ;SET PRIORITY TO 7
  (3)  022300  010400                        MOV     R4,R0
  (3)  022302  104441                        TRAP    C$SPRI
 7321  022304  013705  002700                MOV     STAT1,R5        ;GET BR LEVEL OF M8200,4,7
```

```
 7322  022310  006205                        ASR     R5              ;SHIFT R5 4 TIMES
 7323  022312  006205                        ASR     R5              ;TO GET PROPER LEVEL
 7324  022314  006205                        ASR     R5
 7325  022316  006205                        ASR     R5
 7326  022320  042705  177437                BIC     #177437,R5      ;CLEAR UNWANTED BITS
 7327  022324  010502                        MOV     R5,R2           ;PUT M8200,4,7 LEVEL IN R2
 7328  022326  162702  000040                SUB     #40,R2          ;GET NEXT LOWER LEVEL IN R2
 7329  022332  004537  003536                JSR     R5,SETVEC       ;SET UP VECTORS
 7330  022336  022420                        2$                      ;A VECTOR
 7331  022340  022426                        3$                      ;B VECTOR
 7332  022342  000340  000340                .WORD   340,340         ;PRIORITY 7
 7333  022346  012761  000200  000004  4$:   MOV     #200,4(R1)      ;LOAD PORT4
 7334  022354                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
   (1) 022354  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7335  022360  121111                        121111                  ;SET BR REQUEST
 7336  022362                          5$:   SETPRI  R2              ;PUT LEVEL IN R2 IN PS
   (3) 022362  010200                        MOV     R2,R0
   (3) 022364  104441                        TRAP    C$SPRI
 7337  022366  000240                        NOP
 7338  022370                                ERROR   31              ;ERROR, NO INTERRUPT
   (5) 022406  104455                        TRAP    C$ERDF
   (6) 022410  000037                        .WORD   31
   (6) 022412  005276                        .WORD   EM31
   (6) 022414  010130                        .WORD   ERR31
 7339  022416  000421                  6$:   BR      1$
 7340  022420  012716  022416          2$:   MOV     #6$,(SP)        ;SET UP FOR RTI
 7341  022424  000002                        RTI
 7342  022426                          3$:   ERROR   32              ;ERROR, WRONG VECTOR
   (5) 022444  104455                        TRAP    C$ERDF
   (6) 022446  000040                        .WORD   32
   (6) 022450  005325                        .WORD   EM32
   (6) 022452  010156                        .WORD   ERR32
 7343  022454  012716  022462                MOV     #1$,(SP)        ;SET UP FOR RTI
 7344  022460  000002                        RTI
 7345  022462                          1$:   MSTCLR
   (1) 022462  004537  003142                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 7346  022466                                ENDTST
   (3) 022466                        L10115:
   (3) 022466  104401                        TRAP    C$ETST
 7347
 7348  022470                                BADHEAD
   (2)                                     ;******************** TEST 36 ********************
 7349                                      ;*NPR TEST
 7350                                      ;*TEST OF DATO, 1 WORD FROM UPROC TO 11 MEMORY
 7351  022470                              BADHEAD
   (2)                                     ;******************** TEST 36 ********************
 7352
 7353  022470                        BGNTST
   (3) 022470                        T36::
 7354  022470                                BRESET                  ;BUS RESET
   (3) 022470  104433                        TRAP    C$RESET
 7355
 7356  022472                                MYINT
   (1) 022472  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7357  022476  005011                        CLR     (R1)            ;CLEAR RUN
 7358  022500  005061  000004                CLR     4(R1)           ;CLR PORT4
```

```
7359  022504  004537  003560            JSR     R5,NPRSET       ;SET UP IBUS REG 0-7
7360  022510  000000                    0                       ;IN DATA
7361  022512  177777                    -1                      ;OUT DATA
7362  022514  022630                    3$                      ;IN BA
7363  022516  022626                    2$                      ;OUT BA
7364  022520  005037  022626            CLR     2$              ;CLEAR 2$
7365  022524  005061  000004            CLR     4(R1)           ;CLEAR PORT 4
7366  022530                            ROMCLK                  ;NOW MOVE TO IBUS*<11>
 (1)  022530  004537  003230            JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7367  022534  121111                    121111
7368  022536  012761  000021  000004    MOV     #21,4(R1)       ;WRITE PORT4
7369  022544                            ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  022544  004537  003230            JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7370  022550  121110                    121110                  ;SET NPR BITS IN IBUS* REG 10
7371  022552  000240                    NOP
7372  022554  012737  177777  002636    MOV     #-1,$GDDAT              ;PUT "EXPECTED" IN $GDDAT
7373  022562  013704  022626            MOV     2$,R4           ;PUT "FOUND" IN R4
7374  022566  023704  002636            CMP     $GDDAT,R4               ;DATA CORRECT?
7375  022572  001413                    BEQ     4$              ;BR IF YES
7376  022574                            ERROR   11,YES          ;ERROR NPR FAILED
 (5)  022606  104455                    TRAP    C$ERDF
 (6)  022610  000013                    .WORD   11
 (6)  022612  004640                    .WORD   EM11
 (6)  022614  006644                    .WORD   ERR11
7377  022616                            ESCAPE  TST
 (3)  022616  104410                    TRAP    C$ESCAPE
 (3)  022620  000012                    .WORD   L10116-.
7378  022622                    4$:     EXIT    TST
 (3)  022622  104432                    TRAP    C$EXIT
 (3)  022624  000006                    .WORD   L10116-.
7379  022626  000000            2$:     0                       ;OUT BA
7380  022630  000000            3$:     0                       ;IN BA
7381  022632                    ENDTST
 (3)  022632                    L10116:
 (3)  022632  104401                    TRAP    C$ETST
7382
7383  022634                            BADHEAD
 (2)                                    ;******************** TEST 37 ********************
7384                                    ;*NPR TEST
7385                                    ;*TEST OF DATI, 1 WORD FROM 11 MEMORY TO UPROC
7386  022634                            BADHEAD
 (2)                                    ;******************** TEST 37 ********************
7387
7388  022634                    BGNTST
 (3)  022634                    T37::
7389  022634                            MYINT
 (1)  022634  013701  002716            MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7390  022640                            MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  022640  004537  003142            JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
7391  022644  005061  000004            CLR     4(R1)           ;CLR PORT4
7392  022650  004537  003560            JSR     R5,NPRSET       ;SET UP IBUS REG 0-7
7393  022654  000000                    0                       ;IN DATA
7394  022656  177777                    -1                      ;OUT DATA
7395  022660  023000                    3$                      ;IN BA
7396  022662  022776                    2$                      ;OUT BA
7397  022664  012737  177777  023000    MOV     #-1,3$          ;PUT DATA IN 3$
```

```
 7398  022672  012761  000001  000004          MOV     #1,4(R1)        ;WRITE PORT4
 7399  022700                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  022700  004537  003230                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7400  022704  121110                          121110                  ;SET NPR BITS IN IBUS* REG 11
 7401  022706  000240                          NOP
 7402  022710  012737  177777  002636          MOV     #-1,$GDDAT              ;PUT ''EXPECTED'' IN $GDDAT
 7403  022716                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  022716  004537  003230                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7404  022722  021004                          021004                  ;MOVE IN DATA LOW BYTE TO PORT4
 7405  022724                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  022724  004537  003230                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7406  022730  021025                          021025                  ;MOVE IN DATA HIGH BYTE TO PORT5
 7407  022732  016104  000004                  MOV     4(R1),R4        ;PUT 'FOUND'' IN R4
 7408  022736  023704  002636                  CMP     $GDDAT,R4               ;DATA CORRECT?
 7409  022742  001413                          BEQ     4$              ;BR IF YES
 7410  022744                                  ERROR   11,YES          ;ERROR NPR FAILED
  (5)  022756  104455                          TRAP    C$ERDF
  (6)  022760  000013                          .WORD   11
  (6)  022762  004640                          .WORD   EM11
  (6)  022764  006644                          .WORD   ERR11
 7411  022766                                  ESCAPE  TST
  (3)  022766  104410                          TRAP    C$ESCAPE
  (3)  022770  000012                          .WORD   L10117-.
 7412  022772                          4$:     EXIT    TST
  (3)  022772  104432                          TRAP    C$EXIT
  (3)  022774  000006                          .WORD   L10117-.
 7413  022776  000000                  2$:     0                       ;OUT BA
 7414  023000  000000                  3$:     0                       ;IN BA
 7415  023002                          ENDTST
  (3)  023002                          L10117:
  (3)  023002  104401                          TRAP    C$ETST
 7416
 7417  023004                                  BADHEAD
  (2)                                         ;******************** TEST 38 ********************
 7418                                         ;*NPR TEST
 7419                                         ;*TEST OF DATOB, 1 BYTE FROM UPROC TO 11 MEMORY
 7420  023004                                  BADHEAD
  (2)                                         ;******************** TEST 38 ********************
 7421
 7422  023004                  BGNTST
  (3)  023004                  T38::
 7423  023004                                  MYINT
  (1)  023004  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7424  023010                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  023010  004537  003142                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 7425  023014  005061  000004                  CLR     4(R1)           ;CLR PORT4
 7426  023020  004537  003560                  JSR     R5,NPRSET       ;SET UP IBUS REG 0-7
 7427  023024  000000                          0                       ;IN DATA
 7428  023026  177777                          -1                      ;OUT DATA
 7429  023030  023144                          3$                      ;IN BA
 7430  023032  023143                          2$+1                    ;OUT BA
 7431  023034  005037  023142                  CLR     2$              ;CLEAR 2$
 7432  023040  005061  000004                  CLR     4(R1)           ;CLEAR PORT 4
 7433  023044                                  ROMCLK                  ;NOW MOVE IT TO IBUS*<11>
  (1)  023044  004537  003230                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7434  023050  121111                          121111
```

```
 7435  023052  012761  000221  000004          MOV     #221,4(R1)      ;WRITE PORT4
 7436  023060                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  023060  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7437  023064  121110                          121110                  ;SET NPR BITS IN IBUS* REG 11
 7438  023066  000240                          NOP
 7439  023070  012737  177400  002636          MOV     #177400,$GDDAT  ;PUT 'EXPECTED'' IN $GDDAT
 7440  023076  013704  023142                  MOV     2$,R4           ;PUT 'FOUND'' IN R4
 7441  023102  023704  002636                  CMP     $GDDAT,R4               ;DATA CORRECT?
 7442  023106  001413                          BEQ     4$              ;BR IF YES
 7443  023110                                  ERROR   11,YES          ;ERROR NPR FAILED
  (5)  023122  104455                          TRAP    C$ERDF
  (6)  023124  000013                          .WORD   11
  (6)  023126  004640                          .WORD   EM11
  (6)  023130  006644                          .WORD   ERR11
 7444  023132                                  ESCAPE  TST
  (3)  023132  104410                          TRAP    C$ESCAPE
  (3)  023134  000012                          .WORD   L10120-.
 7445  023136                          4$:     EXIT    TST
  (3)  023136  104432                          TRAP    C$EXIT
  (3)  023140  000006                          .WORD   L10120-.
 7446  023142  000000                  2$:     0                       ;OUT BA
 7447  023144  000000                  3$:     0                       ;IN BA
 7448  023146                          ENDTST
  (3)  023146                          L10120:
  (3)  023146  104401                          TRAP    C$ETST
 7449
 7450  023150                                  BADHEAD
  (2)                                         ;******************** TEST 39 ********************
 7451                                         ;*TEST OF EA BITS 16 AND 17
 7452                                         ;*DO A DATO TO AN ADDRESS USING OUT BA BITS 16 AND 17
 7453                                         ;*VERIFY CORRECT RESULTS
 7454  023150                                  BADHEAD
  (2)                                         ;******************** TEST 39 ********************
 7455
 7456  023150                  BGNTST
  (3)  023150                  T39::
 7457  023150                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  023150  004537  003142                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 7458  023154                                  MYINT
  (1)  023154  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7459  023160  013737  002726  023206          MOV     KMP06,1$        ;USE SEL4 FOR ADDRESS
 7460  023166  013737  002726  023204          MOV     KMP06,2$        ;USE SEL4 FOR ADDRESS
 7461  023174  004537  003560                  JSR     R5,NPRSET       ;LOAD BA AND DATA
 7462  023200  000000                          0                       ;IN DATA
 7463  023202  125252                          125252                  ;OUT DATA
 7464  023204  000000                  2$:     0                       ;IN BA
 7465  023206  000000                  1$:     0                       ;OUT BA
 7466  023210  012761  000014  000004          MOV     #14,4(R1)       ;LOAD SEL 4 WITH OUT BA16 AND 17
 7467  023216                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  023216  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7468  023222  121111                          121111                  ;SET OUTBA 16 AND 17
 7469  023224  012761  000021  000004          MOV     #21,4(R1)       ;LOAD SEL4
 7470  023232  012711  003000                  MOV     #BIT9!BIT10,(R1)
 7471  023236  012761  121110  000006          MOV     #121110,6(R1)   ;PUT INSTRUCTION IN SEL6
 7472  023244  052711  000400                  BIS     #BIT8,(R1)      ;CLOCK IT.
 7473  023250  000240                          NOP                     ;WAIT FOR NPR
```

```
 7474  023252  012737  121110  002636          MOV     #121110,$GDDAT  ;PUT 'EXPECTED'' IN $GDDAT
 7475  023260  000240                          NOP
 7476  023262  000240                          NOP
 7477                                                                  ;OK,LISTEN UP.EXPLAINATION TIME.
 7478                                                                  ;
 7479                                                                  ;ON THE NPR OUT,THE DATA ENDED UP
 7480                                                                  ;IN THE IBUS(NOT IBUS*) SENCE SEL A
 7481                                                                  ;WAS ONLY SELECTED IN THE NPR CYCLE.
 7482                                                                  ;THAT IS,WE DIDN'T REALLY DO AN NPR TO
 7483                                                                  ;PORT 6,THE NPR OUT REALLY ENDED UP IN
 7484                                                                  ;OUT DATA LOW,AND OUT DATA HIGH
 7485                                                                  ;(IBUS <2> AND IBUS <3>).
 7486
 7487                                                                  ;WHAT WE'RE DOING NEXT IS READING IBUS 2&3
 7488                                                                  ;TO SEE IF THE DATA GOT XFERRED CORRECTLY.
 7489  023264                                  ROMCLK
  (1)  023264  004537  003230                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7490  023270  021044                          021044                  ;READ IBUS <2> PUT IN PORT 4
 7491  023272                                  ROMCLK
  (1)  023272  004537  003230                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7492  023276  021065                          021065                  ;READ IBUS <3> PUT IN PORT 5
 7493  023300  016104  000004                  MOV     4(R1),R4        ;PUT 'FOUND'' IN R4
 7494  023304  023704  002636                  CMP     $GDDAT,R4               ;CORRECT RESULTS?
 7495  023310  001411                          BEQ     3$              ;BR IF YES
 7496  023312                                  ERROR   11,YES          ;ERROR BA 16 AND 17 FAILED
  (5)  023324  104455                          TRAP    C$ERDF
  (6)  023326  000013                          .WORD   11
  (6)  023330  004640                          .WORD   EM11
  (6)  023332  006644                          .WORD   ERR11
 7497  023334                          3$:
 7498  023334                          ENDTST
  (3)  023334                          L10121:
  (3)  023334  104401                          TRAP    C$ETST
 7499
 7500  023336                                  BADHEAD
  (2)                                         ;******************** TEST 40 ********************
 7501                                         ;*TEST OF EA BITS 16 AND 17
 7502                                         ;*DO A DATI USING IN BA BITS 16 AND 17
 7503                                         ;*VERIFY CORRECT RESULTS
 7504                                         ;*IN ORDER TO DO THIS TEST, WE WILL READ THE DATA FROM THE
 7505                                         ;*CONSOL TTY CSR IF ONE EXSITS
 7506                                         ;*IF NO CONSOL TTY CSR AT ADDRESS 177560, THIS TEST
 7507                                         ;*WILL BE SKIPPED
 7508  023336                                  BADHEAD
  (2)                                         ;******************** TEST 40 ********************
 7509
 7510  023336                          BGNTST
  (3)  023336                          T40::
 7511  023336                                  MYINT
  (1)  023336  013701  002716                  MOV     KMCSR,R1                ;GET DEVICE ADDRESS.
 7512  023342                                  MSTCLR                          ;MASTER CLEAR M8200,4,7
  (1)  023342  004537  003142                  JSR     R5,.MSTCLR                      ;CLEAR M8200,4,7
 7513  023346  012737  023530  000004  STOP:  MOV     #TOUTT,4                ;SET UP FOR TRAP IN CASE IF NO
 7514  023354  012737  000340  000006          MOV     #340,6  ;TTY AT ADDRESS 177560
 7515  023362  005737  177560                  TST     177560                  ;ADDRESS THE TTY-TRAPS HERE IF NONE.
 7516  023366  012737  177560  023414          MOV     #177560,1$              ;JSE SEL4 FOR ADDRESS
```

```
 7517  023374  012737  177560  023412          MOV     #177560,2$      ;USE SEL4 FOR ADDRESS
 7518  023402  004537  003560                  JSR     R5,NPRSET       ;LOAD BA AND DATA
 7519  023406  000000                          0                       ;IN DATA
 7520  023410  125252                          125252                  ;OUT DATA
 7521  023412  000000                  2$:     0                       ;IN BA
 7522  023414  000000                  1$:     0                       ;OUT BA
 7523  023416  012761  000015  000004          MOV     #15,4(R1)
 7524  023424  012711  003000                  MOV     #BIT9.BIT10,(R1);SET CROMI AND CROMO.!
 7525  023430  012761  121110  000006          MOV     #121110,6(R1)   ;PUT INSTR INTO SEL6  NW*
 7526  023436  052711  000400                  BIS     #BIT8,(R1)      ;CLOCK IT!
 7527  023442  000240                          NOP                     ;WAIT FOR NPR
 7528  023444                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  023444  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7529  023450  021004                          021004                  ;MOVE OUT DATA LB TO SEL4
 7530  023452                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  023452  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7531  023456  021025                          021025                  ;MOVE OUT DATA HB TO SEL5
 7532  023460  016104  000004                  MOV     4(R1),R4        ;PUT 'FOUND' IN R4
 7533  023464  013737  177560  002636          MOV     177560,$GDDAT
 7534  023472  042737  000200  002636          BIC     #200,$GDDAT
 7535  023500  023704  002636                  CMP     $GDDAT,R4               ;CORRECT RESULTS?
 7536  023504  001413                          BEQ     TOUTP           ;BR IF YES
 7537  023506                                  ERROR   11,YES          ;ERROR BA 16 AND 17 FAILED
  (5)  023520  104455                          TRAP    C$ERDF
  (6)  023522  000013                          .WORD   11
  (6)  023524  004640                          .WORD   EM11
  (6)  023526  006644                          .WORD   ERR11
 7538  023530                          3$:
 7539  023530  062706  000004          TOUTT:  ADD     #4,SP           ;UPDATE STACK POITNTER
 7540  023534  013737  002652  000006  TOUTP:  MOV     SAVE6,6         ;RESTORE TRAP VECTOR
 7541  023542  013737  002650  000004          MOV     SAVE4,4
 7542  023550                          ENDTST
  (3)  023550                          L10^22:
  (3)  023550  104401                          TRAP    C$ETST
 7543
 7544  023552                                  BADHEAD
  (2)                                         ;******************** TEST 41 ********************
```

```
 7546                                           ;*NPR NON-EXISTENT MEMORY TEST
 7547                                           ;*DO A DATO TO A NON-EXISTENT ADDRESS
 7548                                           ;*VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
 7549  023552                                   BADHEAD
  (2)                                           ;******************** TEST 41 ********************
 7550                                   /
 7551  023552                                   BGNTST
  (3)  023552                                   T41::
 7552  023552                                           MYINT
  (1)  023552  013701  002716                      MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7553  023556                                    MSTCLR                      ;MASTER CLEAR M8200,4,7
  (1)  023556  004537  003142                      JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 7554  023562  004537  003560                      JSR     R5,NPRSET       ;LOAD IBUS REGISTERS 0-7
 7555  023566  000000                             0                       ;IN DATA
 7556  023570  000000                             0                       ;OUT DATA
 7557  023572  177320                             177320                  ;IN BA
 7558  023574  177320                             177320                  ;IN BA
 7559  023576  012761  000014  000004             MOV     #14,4(R1)       ;SET OUT BA BITS 16+17 IN PORT4
```

```
7561  023604
 (1)  023604  004537  003230                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
                                             JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7562  023610  121111                        121111                  ;SET OUTBA 16 AND 17
7563  023612  012761  000021  000004        MOV     #21,4(R1)       ;SET NPR REQUEST BITS IN PORT4
7564  023620                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  023620  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7565  023624  121110                        121110                  ;MOV IBUS* 4 TO IBUS* 10
7566  023626  000240                        NOP
7567  023630                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  023630  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7568  023634  121225                        121225                  ;MOV IBUS*11 TO IBUS*5
7569  023636  012737  000001  002636        MOV     #1,$GDDAT               ;PUT 'EXPECTED'' IN $GDDAT
7570  023644  116104  000005                MOVB    5(R1),R4        ;PUT 'FOUND'' IN R4
7571  023650  042704  177776                BIC     #177776,R4      ;CLEAR UNWANTED BITS
7572  023654  023704  002636                CMP     $GDDAT,R4               ;DATA CORRECT?
7573  023660  001411                        BEQ     1$              ;BR IF YES
7574  023662                                ERROR   13,YES          ;ERROR NON-EXISTENT MEM BIT FAILED TO SET
 (5)  023674  104455                        TRAP    C$ERDF
 (6)  023676  000015                        .WORD   13
 (6)  023700  004673                        .WORD   EM13
 (6)  023702  007000                        .WORD   ERR13
7575  023704                          1$:
7576  023704  152761  000100  000001        BISB    #100,1(R1)      ;SET MASTER CLEAR
7577  023712  142761  000100  000001        BICB    #100,1(R1)      ;CLEAR MASTER
7578  023720                                ROMCLK                  ;MOV IBUS*11 TO
 (1)  023720  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7579  023724  121225                        121225                  ;PORT5
7580  023726  005037  002636                CLR     $GDDAT          ;EXPECT CLEAR
7581  023732  116104  000005                MOVB    5(R1),R4        ;GET NPR REG
7582  023736  042704  177776                BIC     #177776,R4      ;CLEAR JUNK
7583  023742  001411                        BEQ     2$              ;EXIT IF CLEAR
7584  023744                                ERROR   13,YES          ;NON-EXISTANT MEM
 (5)  023756  104455                        TRAP    C$ERDF
 (6)  023760  000015                        .WORD   13
 (6)  023762  004673                        .WORD   EM13
 (6)  023764  007000                        .WORD   ERR13
7585
7586  023766                          2$:                             ;BIT FAILED TO CLEAR
7587  023766                          ENDTST
 (3)  023766                          L10123:
 (3)  023766  104401                        TRAP    C$ETST
7588
7589  023770                                BADHEAD
 (2)                                        ;******************** TEST 42 ********************
7590                                        ;*NPR NON-EXISTENT MEMORY TEST
7591                                        ;*DO A DATI FROM A NON-EXISTENT ADDRESS
7592                                        ;*VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
7593  023770                                BADHEAD
 (2)                                        ;******************** TEST 42 ********************
7594
7595  023770                          BGNTST
 (3)  023770                          T42::
7596  023770                                MYINT
 (1)  023770  013701  002716                MOV     KMCSR,R1                ;GET DEVICE ADDRESS.
7597  023774                                MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  023774  004537  003142                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
```

```
 7598  024000  004537  003560                JSR     R5,NPRSET       ;LOAD IBUS REGISTERS 0-7
 7599  024004  000000                        0                       ;IN DATA
 7600  024006  000000                        0                       ;OUT DATA
 7601  024010  177320                        177320                  ;IN BA
 7602  024012  177320                        177320                  ;OUT BA
 7603  024014  005061  000004                CLR     4(R1)
 7604  024020                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  024020  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7605  024024  121111                        121111                  ;CLEAR NON-EXISTENT BIT
 7606  024026  012761  000015  000004        MOV     #15,4(R1)       ;SET NPR REQUEST BITS IN PORT4
 7607  024034                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  024034  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7608  024040  121110                        121110                  ;MOV IBUS* 4 TO IBUS* 10
 7609  024042  000240                        NOP
 7610  024044                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  024044  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7611  024050  121225                        121225                  ;MOV IBUS*11 TO IBUS*5
 7612  024052  012737  000001  002636        MOV     #1,$GDDAT               ;PUT 'EXPECTED' IN $GDDAT
 7613  024060  116104  000005                MOVB    5(R1),R4        ;PUT 'FOUND' IN R4
 7614  024064  042704  177776                BIC     #177776,R4      ;CLEAR UNWANTED BITS
 7615  024070  023704  002636                CMP     $GDDAT,R4               ;DATA CORRECT?
 7616  024074  001411                        BEQ     1$              ;BR IF YES
 7617  024076                                ERROR   13,YES          ;ERROR NON-EXISTENT MEM BIT FAILED TO SET
  (5)  024110  104455                        TRAP    C$ERDF
  (6)  024112  000015                        .WORD   13
  (6)  024114  004673                        .WORD   EM13
  (6)  024116  007000                        .WORD   ERR13
 7618  024120                        1$:
 7619  024120                        ENDTST
  (3)  024120                        L10124:
  (3)  024120  104401                        TRAP    C$ETST
 7620
 7621  024122                                BADHEAD
  (2)                                       ;********************* TEST 43 *********************
 7622                                       ;*NPR TEST
 7623                                       ;*USING DATO, NPR A BINARY COUNT (0-377)
 7624                                       ;*FROM MICRO-PROCESSOR TO ALL AVAILABLE MEMORY
 7625  024122                                BADHEAD
  (2)                                       ;********************* TEST 43 *********************
 7626
 7627  024122                        BGNTST
  (3)  024122                        T43::
 7628  024122                                MYINT
  (1)  024122  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7629  024126                                MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  024126  004537  003142                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 7630  024132  005037  024334                CLR     5$              ;START FLAG AT 0
 7631  024136  005005                        CLR     R5              ;DATA
 7632  024140  012702  035450                MOV     #CORMAX,R2      ;ADDRESS
 7633  024144                        1$:
 7634  024144  010537  024174                MOV     R5,2$           ;LOAD DATA
 7635  024150  010237  024200                MOV     R2,4$           ;LOAD BA
 7636  024154  032702  000001                BIT     #BIT0,R2        ;IS BA ODD?
 7637  024160  001402                        BEQ     .+6             ;BR IF NO
 7638  024162  000337  024174                SWAB    2$              ;IF ODD PUT DATA IN HI-BYTE
 7639  024166  004537  003560                JSR     R5,NPRSET       ;LOAD NPR REGISTERS
```

```
7640  024172  000000                        0                       ;IN DATA
7641  024174  000000                2$:     0                       ;OUT DATA
7642  024176  000000                        0                       ;IN BA
7643  024200  000000                4$:     0                       ;OUT BA
7644  024202  105012                        CLRB    (R2)            ;CLEAR MEMORY LOCATION
7645  024204  012761  000221  000004       MOV     #221,4(R1)      ;LOAD PORT4
7646  024212                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  024212  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7647  024216  121110                        121110                  ;DO THE NPR
7648  024220  000240                        NOP
7649  024222  010537  002636                MOV     R5,$GDDAT               ;PUT 'EXPECTED'' IN $GDDAT
7650  024226  111204                        MOVB    (R2),R4 ;PUT 'FOUND'' IN R4
7651  024230  123704  002636                CMPB    $GDDAT,R4               ;IS DATA CORRECT?
7652  024234  001411                        BEQ     3$              ;BR IF YES
7653  024236                                ERROR   11,YES          ;ERROR, DATA INCORRECT
 (5)  024250  104455                        TRAP    C$ERDF
 (6)  024252  000013                        .WORD   11
 (6)  024254  004640                        .WORD   EM11
 (6)  024256  006644                        .WORD   ERR11
7654  024260                        3$:     ESCAPE  TST
 (3)  024260  104410                        TRAP    C$ESCAPE
 (3)  024262  000054                        .WORD   L10125-.
7655  024264  005205                        INC     R5              ;NEXT CHARACTER
7656  024266  042705  177400                BIC     #177400,R5      ;USE ONLY LOW BYTE
7657  024272  005737  024334                TST     5$              ;HAS MAX MEMORY BEEN REACHED YET?
7658  024276  001402                        BEQ     6$              ;BR IF NO
7659  024300  005705                        TST     R5              ;DONE PATTERN?
7660  024302  001412                        BEQ     7$              ;BR IF YES
7661  024304  005202                6$:     INC     R2              ;INC BA
7662  024306  023702  002604                CMP     MEMLIM,R2       ;REACHED MEMORY LIMIT YET?
7663  024312  001314                        BNE     1$              ;BR IF NOT
7664  024314  012702  035450                MOV     #CORMAX,R2      ;RESTART BA AT FIRST ADDRESS
7665  024320  012737  177777  024334        MOV     #-1,5$          ;SET FLAG TO END TEST AT END OF DATA PATTERN
7666  024326  000706                        BR 1$                   ;CONTINUE
7667  024330                        7$:
7668  024330                                EXIT    TST
 (3)  024330  104432                        TRAP    C$EXIT
 (3)  024332  000004                        .WORD   L10125-.
7669  024334  000000                5$:     0                       ;THIS LOCATION IS A FLAG, IT STARTS AT 0,
7670                                                                ;AND IS SET TO -1 WHEN LAST MEMORY ADDRESS
7671                                                                ;IS USED, TEST IS THEN ENDED WHEN PATTERN IS FINISHED
7672  024336                        ENDTST
 (3)  024336                        L10125:
 (3)  024336  104401                        TRAP    C$ETST
7673                                ;$MEM1
7674                                ;$MEM0
7675                                ;$MEM2  1K
7676                                ;$MEM3  1K
7677
7678  024340                                BADHEAD
 (2)                                        ;******************** TEST 44 ********************
7679                                        ;*ALU C BIT TEST
7680                                        ;*TEST THAT AN ADD OF 377 AND 377 WILL SET THE C BIT
7681  024340                                BADHEAD
 (2)                                        ;******************** TEST 44 ********************
7682
```

```
 7683  024340                                  BGNTST
  (3)  024340                                  T44::
 7684  024340                                          MYINT
  (1)  024340  013701  002716                         MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7685  024344                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  024344  004537  003142                         JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 7686  024350  004737  003624                         JSR     PC,MEMLD        ;LOAD MAINMEM DATA
 7687  024354  024470                              TDATA                       ;POINTER TO DATA
 7688  024356  004737  003776                         JSR     PC,SPLD         ;LOAD SP DATA
 7689  024362  024470                              TDATA                       ;POINTER TO DATA
 7690  024364                                          BGNSEG
  (3)  024364  104404                                 TRAP    C$BSEG
 7691  024366                                  1$:
 7692  024366                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  024366  004537  003230                         JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7693  024372  010000                              010000                      ;MAR 0
 7694  024374                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  024374  004537  003230                         JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7695  024400  054400                              054400!<0*20>               ;ADD 377 AND 377, TO SET C BIT
 7696  024402                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  024402  004537  003230                         JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7697  024406  040421                              040401!<1*20>               ;ADD 0 AND 0 AND THE C BIT
 7698  024410                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  024410  004537  003230                         JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7699  024414  061224                              61224                       ;PUT RESULTS IN PORT4
 7700  024416  012737  000001  002636                 MOV     #1,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
 7701  024424  016104  000004                         MOV     4(R1),R4        ;PUT "FOUND" IN R4
 7702  024430  123704  002636                         CMPB    $GDDAT,R4               ;DATA CORRECT?
 7703  024434  001411                                 BEQ     2$              ;BR IF YES
 7704  024436                                          ERROR   34,YES          ;ERROR C BIT NOT SET
  (5)  024450  104455                                 TRAP    C$ERDF
  (6)  024452  000042                                 .WORD   34
  (6)  024454  005421                                 .WORD   EM34
  (6)  024456  010262                                 .WORD   ERR34
 7705  024460                                  2$:     ESCAPE  SEG
  (3)  024460  104410                                 TRAP    C$ESCAPE
  (3)  024462  000002                                 .WORD   10000$-.
 7706  024464                                          ENDSEG
  (3)  024464                                  10000$:
  (3)  024464  104405                                 TRAP    C$ESEG
 7707  024466                                  ENDTST
  (3)  024466                                  L10126:
  (3)  024466  104401                                 TRAP    C$ETST
 7708  024470     377     000     000  TDATA:  .BYTE   -1,0,0,0,0,0,0,0
       024473     000     000     000
       024476     000     000
 7709
 7710                                          .EVEN
 7711
 7712  024500                                          BADHEAD
  (2)                                             ;********************* TEST 45 *********************
 7713                                             ;*ALU TEST
 7714                                             ;*TEST OF ALU FUNCTION SEL B WITH C BIT CLEARED
 7715                                             ;*ALU FUNCTION (B)    CODE=11
 7716                                             ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 7717                                             ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
```

```
                                        BADHEAD
                                        ;******************** TEST 45 ********************

                                BGNTST
                                T45::
                                        MYINT
 (1)  024500  013701  002716            MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  024504  004537  003142            JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
                                        CLR     R5              ;MEM + SP ADDRESS
                                        MOV     #5$,R2          ;POINTER TO CORRECT DATA
                                        JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
                                        MEMDAT                  ;POINTER TO DATA
                                        JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
                                        SPDAT                   ;POINTER TO DATA
                                        BGNSEG
 (3)  024532  104404                    TRAP    C$BSEG
                                1$:     JSR     PC,CLRC         ;CLEAR C BIT!
                                        BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
                                        BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
                                        ROMCLK                  ;NEXT WORD IS INSTPUCTION, ROMCLK PC=5304
 (1)  024552  004537  003230            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
                                2$:     010000                  ;LOAD MAR
                                        BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
                                        BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  024572  004537  003230            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
                                3$:     040400!<11*20>          ;BR   SEL B
                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  024600  004537  003230            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
                                        61224                   ;MOVE BR TO PORT4
                                        MOVB    (R2),$GDDAT             ;PUT 'EXPECTED'' IN $GDDAT
                                        MOVB    4(R1),R4        ;PUT 'FOUND'' IN R4
                                        CMPB    $GDDAT,R4               ;DATA CORRECT?
                                        BEQ     4$              ;BR IF YES
                                        ERROR   15,YES          ;ALU ERROR
                                        TRAP    C$ERDF
                                        .WORD   15
                                        .WORD   EM15
                                        .WORD   ERR15
                                4$:     ESCAPE  SEG
                                        TRAP    C$ESCAPE
                                        .WORD   10000$-.
                                        INC     R2              ;NEXT DATA
                                        INC     R5              ;NEXT ADDRESS
                                        CMP     #10,R5          ;DONE YET?
                                        BNE     1$              ;BR IF NO
                                        ENDSEG
                                10000$:
                                        TRAP    C$ESEG
                                        EXIT    TST
                                        TRAP    C$EXIT
                                        .WORD   L10127-.
                                5$:     .BYTE   0,-1,0,-1,125,252,125,252
```

```
 7754
 7755                                   .EVEN
 7756  024702                           ENDTST
  (3)  024702                           L10127:
  (3)  024702  104401                            TRAP    C$ETST
 7757
 7758  024704                                   BADHEAD
  (2)                                           ;******************** TEST 46 ********************
 7759                                           ;*ALU TEST
 7760                                           ;*TEST OF ALU FUNCTION SEL A WITH C BIT CLEARED
 7761                                           ;*ALU FUNCTION (A)    CODE=10
 7762                                           ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 7763                                           ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 7764  024704                                   BADHEAD
  (2)                                           ;******************** TEST 46 ********************
 7765
 7766  024704                           BGNTST
  (3)  024704                           T46::
 7767  024704                                   MYINT
  (1)  024704  013701  002716                     MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7768  024710                                   MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  024710  004537  003142                     JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 7769  024714  005005                           CLR     R5              ;MEM + SP ADDRESS
 7770  024716  012702  025076                   MOV     #5$,R2          ;POINTER TO CORRECT DATA
 7771  024722  004737  003624                   JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 7772  024726  002654                           MEMDAT                  ;POINTER TO DATA
 7773  024730  004737  003776                   JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 7774  024734  002664                           SPDAT                   ;POINTER TO DATA
 7775  024736                                   BGNSEG
  (3)  024736  104404                           TRAP    C$BSEG
 7776  024740  004737  004044          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
 7777  024744  042737  000017  024762           BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7778  024752  050537  024762                   BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 7779  024756                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
  (1)  024756  004537  003230                     JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7780  024762  010000                  2$:     010000                  ;LOAD MAR
 7781  024764  042737  000017  025002           BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 7782  024772  050537  025002                   BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 7783  024776                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
  (1)  024776  004537  003230                     JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7784  025002  040600                  3$:     040400!<10*20>          ;BR   SEL A
 7785  025004                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  025004  004537  003230                     JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7786  025010  061224                           61224                   ;MOVE BR TO PORT4
 7787  025012  111237  002636                   MOVB    (R2),$GDDAT             ;PUT 'EXPECTED'' IN $GDDAT
 7788  025016  116104  000004                   MOVB    4(R1),R4        ;PUT 'FOUND'' IN R4
 7789  025022  123704  002636                   CMPB    $GDDAT,R4               ;DATA CORRECT?
 7790  025026  001411                           BEQ     4$              ;BR IF YES
 7791  025030                                   ERROR   15,YES          ;ALU ERROR
  (5)  025042  104455                           TRAP    C$ERDF
  (6)  025044  000017                           .WORD   15
  (6)  025046  004740                           .WORD   EM15
  (6)  025050  007104                           .WORD   ERR15
 7792  025052                           '$:     ESCAPE  SEG
  (3)  025052  104410                           TRAP    C$ESCAPE
  (3)  025054  000014                           .WORD   10000$-.
```

```
                                                INC     R2              ;NEXT DATA
                                                INC     R5              ;NEXT DATA
                                                CMP     #10,R5          ;DONE YET?
                                                BNE     1$              ;BR IF NO
                                                ENDSEG
                                        10000$:
                                                TRAP    C$ESEG
                                                EXIT    TST
                                                TRAP    C$EXIT
                                                .WORD   L10130-.
                                        5$:     .BYTE   0,0,-1,-1,125,125,252,252

                                                .EVEN
                                                ENDTST
                                        L10130:
                                                TRAP    C$ETST

                                                BADHEAD
                                                ;******************** TEST 47 ********************
                                                ;*ALU TEST
                                                ;*TEST OF ALU FUNCTION A OR NOTB WITH C BIT CLEARED
                                                ;*ALU FUNCTION (A OR NOTB)    CODE=12
                                                ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
                                                BADHEAD
                                                ;******************** TEST 47 ********************

                                        BGNTST
                                        T47::
                                                MYINT
                                                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
                                                MSTCLR                  ;MASTER CLEAR M8200,4,7
                                                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
                                                CLR     R5              ;MEM + SP ADDRESS
                                                MOV     #5$,R2          ;POINTER TO CORRECT DATA
                                                JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
                                                MEMDAT                  ;POINTER TO DATA
                                                JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
                                                SPDAT                   ;POINTER TO DATA
                                                BGNSEG
                                                TRAP    C$BSEG
                                        1$:     JSR     PC,CLRC         ;CLEAR C BIT!
                                                BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
                                                BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
                                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
                                                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
                                        2$:     010000                  ;LOAD MAR
                                                BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
                                                BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
                                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
                                                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
                                        3$:     040400!<12*20>          ;BR   A OR NOTB
                                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
                                                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
```

```
7832  025214  061224                          61224                   ;MOVE BR TO PORT4
7833  025216  111237  002636                  MOVB    (R2),$GDDAT             ;PUT 'EXPECTED' IN $GDDAT
7834  025222  116104  000004                  MOVB    4(R1),R4        ;PUT 'FOUND' IN R4
7835  025226  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
7836  025232  001411                          BEQ     4$              ;BR IF YES
7837  025234                                  ERROR   15,YES          ;ALU ERROR
 (5)  025246  104455                          TRAP    C$ERDF
 (6)  025250  000017                          .WORD   15
 (6)  025252  004740                          .WORD   EM15
 (6)  025254  007104                          .WORD   ERR15
7838  025256                          4$:     ESCAPE  SEG
 (3)  025256  104410                          TRAP    C$ESCAPE
 (3)  025260  000014                          .WORD   10000$-.
7839  025262  005202                          INC     R2              ;NEXT DATA
7840  025264  005205                          INC     R5              ;NEXT DATA
7841  025266  022705  000010                  CMP     #10,R5          ;DONE YET?
7842  025272  001324                          BNE     1$              ;BR IF NO
7843  025274                                  ENDSEG
 (3)  025274                          10000$:
 (3)  025274  104405                          TRAP    C$ESEG
7844  025276                                  EXIT    TST
 (3)  025276  104432                          TRAP    C$EXIT
 (3)  025300  000012                          .WORD   L10131-.
7845  025302     377      000      377  5$:     .BYTE   -1,0,-1,-1,-1,125,252,-1
      025305     377      377      125
      025310     252      377
7846
7847                                  .EVEN
7848  025312                          ENDTST
 (3)  025312                          L10131:
 (3)  025312  104401                          TRAP    C$ETST
7849
7850  025314                                  BADHEAD
 (2)                                          ;********** TEST 48 **********
7851                                          ;*ALU TEST
7852                                          ;*TEST OF ALU FUNCTION A AND B WITH C BIT CLEARED
7853                                          ;*ALU FUNCTION (A AND B)    CODE=13
7854                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7855                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7856  025314                                  BADHEAD
 (2)                                          ;********** TEST 48 **********
7857
7858  025314                          BGNTST
 (3)  025314                          T48::
7859  025314                                  MYINT
 (1)  025314  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7860  025320                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  025320  004537  003142                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
7861  025324  005005                          CLR     R5              ;MEM + SP ADDRESS
7862  025326  012702  025506                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
7863  025332  004737  003624                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7864  025336  002654                          MEMDAT                  ;POINTER TO DATA
7865  025340  004737  003776                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7866  025344  002664                          SPDAT                   ;POINTER TO DATA
7867  025346                                  BGNSEG
 (3)  025346  104404                          TRAP    C$BSEG
```

```
 7868  025350  004737  004044          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
 7869  025354  042737  000017  025372          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7870  025362  050537  025372                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 7871  025366                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
   (1) 025366  004537  003230                  JSR     R5,,ROMCLK              ;CLOCK INSTRUCTION
 7872  025372  010000                  2$:     010000                  ;LOAD MAR
 7873  025374  042737  000017  025412          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 7874  025402  050537  025412                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 7875  025406                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
   (1) 025406  004537  003230                  JSR     R5,,ROMCLK              ;CLOCK INSTRUCTION
 7876  025412  040660                  3$:     040400.<13*20>          ;BR   A AND B
 7877  025414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
   (1) 025414  004537  003230                  JSR     R5,,ROMCLK              ;CLOCK INSTRUCTION
 7878  025420  061224                          61224                   ;MOVE BR TO PORT4
 7879  025422  111237  002636                  MOVB    (R2),$GDDAT             ;PUT 'EXPECTED' IN $GDDAT
 7880  025426  116104  000004                  MOVB    4(R1),R4        ;PUT 'FOUND' IN R4
 7881  025432  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
 7882  025436  001411                          BEQ     4$              ;BR IF YES
 7883  025440                                  ERROR   15,YES          ;ALU ERROR
   (5) 025452  104455                          TRAP    C$ERDF
   (6) 025454  000017                          .WORD   15
   (6) 025456  004740                          .WORD   EM15
   (6) 025460  007104                          .WORD   ERR15
 7884  025462                          4$:     ESCAPE  SEG
   (3) 025462  104410                          TRAP    C$ESCAPE
   (3) 025464  000014                          .WORD   10000$-.
 7885  025466  005202                          INC     R2              ;NEXT DATA
 7886  025470  005205                          INC     R5              ;NEXT DATA
 7887  025472  022705  000010                  CMP     #10,R5          ;DONE YET?
 7888  025476  001324                          BNE     1$              ;BR IF NO
 7889  025500                                  ENDSEG
   (3) 025500                          10000$:
   (3) 025500  104405                          TRAP    C$ESEG
 7890  025502                                  EXIT    TST
   (3) 025502  104432                          TRAP    C$EXIT
   (3) 025504  000012                          .WORD   L10132-.
 7891  025506     000     000     000  5$:     .BYTE   0,0,0,-1,125,0,0,252
       025511     377     125     000
       025514     000     252
 7892
 7893                                  .EVEN
 7894  025516                          ENDTST
   (3) 025516                          L10132:
   (3) 025516  104401                          TRAP    C$ETST
 7895
 7896  025520                                  BADHEAD
   (2)                                       ;********************* TEST 49 *********************
 7897                                        ;*ALU TEST
 7898                                        ;*TEST OF ALU FUNCTION A OR B WITH C BIT CLEARED
 7899                                        ;*ALU FUNCTION (A OR B)     CODE=14
 7900                                        ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 7901                                        ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 7902  025520                                  BADHEAD
   (2)                                       ;********************* TEST 49 *********************
 7903
 7904  025520                          BGNTST
```

```
   (3)  025520                               T49::
 7905   025520                                        MYINT
   (1)  025520  013701  002716                        MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7906   025524                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
   (1)  025524  004537  003142                        JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 7907   025530  005005                                CLR     R5              ;MEM + SP ADDRESS
 7908   025532  012702  025712                        MOV     #5$,R2          ;POINTER TO CORRECT DATA
 7909   025536  004737  003624                        JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 7910   025542  002654                                MEMDAT                  ;POINTER TO DATA
 7911   025544  004737  003776                        JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 7912   025550  002664                                SPDAT                   ;POINTER TO DATA
 7913   025552                                        BGNSEG
   (3)  025552  104404                                TRAP    C$BSEG
 7914   025554  004737  004044           1$:          JSR     PC,CLRC         ;CLEAR C BIT!
 7915   025560  042737  000017  025576                BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7916   025566  050537  025576                        BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 7917   025572                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
   (1)  025572  004537  003230                        JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7918   025576  010000                   2$:          010000                  ;LOAD MAR
 7919   025600  042737  000017  025616                BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 7920   025606  050537  025616                        BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 7921   025612                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
   (1)  025612  004537  003230                        JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7922   025616  040700                   3$:          040400!<14*20>          ;BR   A OR B
 7923   025620                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
   (1)  025620  004537  003230                        JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7924   025624  061224                                61224                   ;MOVE BR TO PORT4
 7925   025626  111237  002636                        MOVB    (R2),$GDDAT             ;PUT 'EXPECTED'' IN $GDDAT
 7926   025632  116104  000004                        MOVB    4(R1),R4        ;PUT 'FOUND'' IN R4
 7927   025636  123704  002636                        CMPB    $GDDAT,R4               ;DATA CORRECT?
 7928   025642  001411                                BEQ     4$              ;BR IF YES
 7929   025644                                        ERROR   15,YES          ;ALU ERROR
   (5)  025656  104455                                TRAP    C$ERDF
   (6)  025660  000017                                .WORD   15
   (6)  025662  004740                                .WORD   EM15
   (6)  025664  007104                                .WORD   ERR15
 7930   025666                           4$:          ESCAPE  SEG
   (3)  025666  104410                                TRAP    C$ESCAPE
   (3)  025670  000014                                .WORD   10000$-.
 7931   025672  005202                                INC     R2              ;NEXT DATA
 7932   025674  005205                                INC     R5              ;NEXT DATA
 7933   025676  022705  000010                        CMP     #10,R5          ;DONE YET?
 7934   025702  001324                                BNE     1$              ;BR IF NO
 7935   025704                                        ENDSEG
   (3)  025704                           10000$:
   (3)  025704  104405                                TRAP    C$ESEG
 7936   025706                                        EXIT    TST
   (3)  025706  104432                                TRAP    C$EXIT
   (3)  025710  000012                                .WORD   L10133-.
 7937   025712     000     377     377   5$:          .BYTE   0,-1,-1,-1,125,-1,-1,252
        025715     377     125     377
        025720     377     252
 7938
 7939                                    .EVEN
 7940   025722                           ENDTST
   (3)  025722                           L10133:
```

```
  (3)  025722  104401                          TRAP    C$ETST
7941
7942  025724                                   BADHEAD
  (2)                                          ;********************* TEST 50 *********************
7943                                           ;*ALU TEST
7944                                           ;*TEST OF ALU FUNCTION A XOR B WITH C BIT CLEARED
7945                                           ;*ALU FUNCTION (A XOR B)     CODE=15
7946                                           ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7947                                           ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7948  025724                                   BADHEAD
  (2)                                          ;********************* TEST 50 *********************
7949
7950  025724                          BGNTST
  (3)  025724                          T50::
7951  025724                                   MYINT
  (1)  025724  013701  002716                 MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7952  025730                                   MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  025730  004537  003142                 JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
7953  025734  005005                          CLR     R5              ;MEM + SP ADDRESS
7954  025736  012702  026116                 MOV     #5$,R2          ;POINTER TO CORRECT DATA
7955  025742  004737  003624                 JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7956  025746  002654                          MEMDAT                  ;POINTER TO DATA
7957  025750  004737  003776                 JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7958  025754  002664                          SPDAT                   ;POINTER TO DATA
7959  025756                                   BGNSEG
  (3)  025756  104404                        TRAP    C$BSEG
7960  025760  004737  004044          1$:    JSR     PC,CLRC         ;CLEAR C BIT!
7961  025764  042737  000017  026002         BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7962  025772  050537  026002                 BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
7963  025776                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  025776  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7964  026002  010000                  2$:    010000                  ;LOAD MAR
7965  026004  042737  000017  026022         BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7966  026012  050537  026022                 BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
7967  026016                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  026016  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7968  026022  040720                  3$:    040400!<15*20>          ;BR   A XOR B
7969  026024                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  026024  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION!
7970  026030  061224                          61224                   ;MOVE BR TO PORT4
7971  026032  111237  002636                 MOVB    (R2),$GDDAT             ;PUT ''EXPECTED'' IN $GDDAT
7972  026036  116104  000004                 MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
7973  026042  123704  002636                 CMPB    $GDDAT,R4               ;DATA CORRECT?
7974  026046  001411                          BEQ     4$              ;BR IF YES
7975  026050                                   ERROR   15,YES          ;ALU ERROR
  (5)  026062  104455                        TRAP    C$ERDF
  (6)  026064  000017                        .WORD   15
  (6)  026066  004740                        .WORD   EM15
  (6)  026070  007104                        .WORD   ERR15
7976  026072                           4$:    ESCAPE  SEG
  (3)  026072  104410                        TRAP    C$ESCAPE
  (3)  026074  000014                        .WORD   10000$-.
7977  026076  005202                          INC     R2              ;NEXT DATA
7978  026100  005205                          INC     R5              ;NEXT DATA
7979  026102  022705  000010                 CMP     #10,R5          ;DONE YET?
7980  026106  001324                          BNE     1$              ;BR IF NO
```

```
        026110                                          ENDSEG
  (3)   026110                                  10000$:
  (3)   026110  104405                                  TRAP    C$ESEG
        026112                                          EXIT    TST
  (3)   026112  104432                                  TRAP    C$EXIT
  (3)   026114  000012                                  .WORD   L10134-.
        026116     000     377     377  5$:             .BYTE   0,-1,-1,0,0,-1,-1,0
        026121     000     000     377
        026124     377     000

                                                .EVEN
        026126                                  ENDTST
  (3)   026126                                  L10134:
  (3)   026126  104401                                  TRAP    C$ETST

        026130                                          BADHEAD
  (2)                                                   ;******************* TEST 51 *******************
                                                        ;*ALU TEST
                                                        ;*TEST OF ALU FUNCTION ADD WITH C BIT CLEARED
                                                        ;*ALU FUNCTION (A PLUS B)     CODE=00
                                                        ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                                        ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
        026130                                          BADHEAD
  (2)                                                   ;******************* TEST 51 *******************

        026130                                  BGNTST
  (3)   026130                                  T51::
        026130                                          MYINT
  (1)   026130  013701  002716                          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
        026134                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)   026134  004537  003142                          JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
        026140  005005                                  CLR     R5              ;MEM + SP ADDRESS
        026142  012702  026322                          MOV     #5$,R2          ;POINTER TO CORRECT DATA
        026146  004737  003624                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
        026152  002654                                  MEMDAT                  ;POINTER TO DATA
        026154  004737  003776                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
        026160  002664                                  SPDAT                   ;POINTER TO DATA
        026162                                          BGNSEG
  (3)   026162  104404                                  TRAP    C$BSEG
        026164  004737  004044          1$:             JSR     PC,CLRC         ;CLEAR C BIT!
        026170  042737  000017  026206                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
        026176  050537  026206                          BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
        026202                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)   026202  004537  003230                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
        026206  010000                  2$:             010000                  ;LOAD MAR
        026210  042737  000017  026226                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
        026216  050537  026226                          BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
        026222                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)   026222  004537  003230                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
        026226  040400                  3$:             040400!<00*20>          ;BR    ADD
        026230                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)   026230  004537  003230                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
        026234  061224                                  61224                   ;MOVE BR TO PORT4
        026236  111237  002636                          MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
        026242  116104  000004                          MOVB    4(R1),R4        ;PUT "FOUND" IN R4
        026246  123704  002636                          CMPB    $GDDAT,R4               ;DATA CORRECT?
```

```
 8020  026252  001411                          BEQ     4$              ;BR IF YES
 8021  026254                                  ERROR   15,YES          ;ALU ERROR
  (5)  026266  104455                          TRAP    C$ERDF
  (6)  026270  000017                          .WORD   15
  (6)  026272  004740                          .WORD   EM15
  (6)  026274  007104                          .WORD   ERR15
 8022  026276                              4$:  ESCAPE  SEG
  (3)  026276  104410                          TRAP    C$ESCAPE
  (3)  026300  000014                          .WORD   10000$-.
 8023  026302  005202                          INC     R2              ;NEXT DATA
 8024  026304  005205                          INC     R5              ;NEXT DATA
 8025  026306  022705  000010                  CMP     #10,R5          ;DONE YET?
 8026  026312  001324                          BNE     1$              ;BR IF NO
 8027  026314                                  ENDSEG
  (3)  026314                          10000$:
  (3)  026314  104405                          TRAP    C$ESEG
 8028  026316                                  EXIT    TST
  (3)  026316  104432                          TRAP    C$EXIT
  (3)  026320  000012                          .WORD   L10135-.
 8029  026322     000      377      377  5$:   .BYTE   0,-1,-1,376,252,-1,-1,124
       026325     376      252      377
       026330     377      124
 8030
 8031                                         .EVEN
 8032  026332                                 ENDTST
  (3)  026332                          L10135:
  (3)  026332  104401                          TRAP    C$ETST
 8033
 8034  026334                                 BADHEAD
  (2)                                         ;****************** TEST 52 ******************
 8035                                         ;*ALU TEST
 8036                                         ;*TEST OF ALU FUNCTION 2A W.C WITH C BIT CLEARED
 8037                                         ;*ALU FUNCTION (A PLUS A PLUS C)        CODE=6
 8038                                         ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 8039                                         ;*PEPFORM THE FUNCTION, VERIFY THE RESULTS
 8040  026334                                 BADHEAD
  (2)                                         ;****************** TEST 52 ******************
 8041
 8042  026334                          BGNTST
  (3)  026334                          T52::
 8043  026334                                  MYINT
  (1)  026334  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 8044  026340                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  026340  004537  003142                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 8045  026344  005005                          CLR     R5              ;MFM + SP ADDRESS
 8046  026346  012702  026526                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
 8047  026352  004737  003624                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 8048  026356  002654                          MEMDAT                  ;POINTER TO DATA
 8049  026360  004737  003776                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 8050  026364  002664                          SPDAT                   ;POINTER TO DATA
 8051  026366                                  BGNSEG
  (3)  026366  104404                          TRAP    C$BSEG
 8052  026370  004737  004044              1$: JSR     PC,CLRC         ;CLEAR C BIT!
 8053  026374  042737  000017  026412         BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 8054  026402  050537  026412                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 8055  026406                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
```

```
  (1)  026406  004537  003230                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
8056   026412  010000                2$:         010000                          ;LOAD MAR
8057   026414  042737  000017  026432            BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
8058   026422  050537  026432                    BIS     R5,3$                   ;ADD ADDRESS TO INSTRUCTION
8059   026426                                    ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  026426  004537  003230                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
8060   026432  040540                3$:         040400!<6*20>                   ;BR   2A W/C
8061   026434                                    ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  026434  004537  003230                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
8062   026440  061224                            61224                           ;MOVE BSR TO PORT4
8063   026442  111237  002636                    MOVB    (R2),$GDDAT             ;PUT 'EXPECTED' IN $GDDAT
8064   026446  116104  000004                    MOVB    4(R1),R4                ;PUT 'FOUND' IN R4
8065   026452  123704  002636                    CMPB    $GDDAT,R4               ;DATA CORRECT?
8066   026456  001411                            BEQ     4$                      ;BR IF YES
8067   026460                                    ERROR   15,YES                  ;ALU ERROR
  (5)  026472  104455                            TRAP    C$ERDF
  (6)  026474  000017                            .WORD   15
  (6)  026476  004740                            .WORD   EM15
  (6)  026500  007104                            .WORD   ERR15
8068   026502                            4$:     ESCAPE  SEG
  (3)  026502  104410                            TRAP    C$ESCAPE
  (3)  026504  000014                            .WORD   10000$-.
8069   026506  005202                            INC     R2                      ;NEXT DATA
8070   026510  005205                            INC     R5                      ;NEXT ADDRESS
8071   026512  022705  000010                    CMP     #10,R5                  ;DONE YET?
8072   026516  001324                            BNE     1$                      ;BR IF NO
8073   026520                                    ENDSEG
  (3)  026520                            10000$:
  (3)  026520  104405                            TRAP    C$ESEG
8074   026522                                    EXIT    TST
  (3)  026522  104432                            TRAP    C$EXIT
  (3)  026524  000012                            .WORD   L10136-.
8075   026526     000      000     376  5$:      .BYTE   0,0,376,376,252,252,124,124
       026531     376      252     252
       026534     124      124
8076
8077                                      .EVEN
8078   026536                             ENDTST
  (3)  026536                             L10136:
  (3)  026536  104401                            TRAP    C$ETST
8079
8080   026540                                    BADHEAD
  (2)                                             ;********** TEST 53 **********
8081                                              ;*ALU TEST
8082                                              ;*TEST OF ALU FUNCTION SUB WITH C BIT CLEARED
8083                                              ;*ALU FUNCTION (A-B)    CODE=16
8084                                              ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8085                                              ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
8086   026540                                    BADHEAD
  (2)                                             ;********** TEST 53 **********
8087
8088   026540                             BGNTST
  (3)  026540                             T53::
8089   026540                                    MYINT
  (1)  026540  013701  002716                    MOV     KMCSR,R1                ;GET DEVICE ADDRESS.
8090   026544                                    MSTCLR                          ;MASTER CLEAR M8200,4,7
```

```
  (1)  026544  004537  003142                    JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
8091   026550  005005                            CLR     R5                      ;MEM + SP ADDRESS
8092   026552  012702  026734                    MOV     #5$,R2                  ;POINTER TO CORRECT DATA
8093   026556  004737  003624                    JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
8094   026562  002654                            MEMDAT                          ;POINTER TO DATA
8095   026564  004737  003776                    JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
8096   026570  002664                            SPDAT                           ;POINTER TO DATA
8097   026572                                    BGNSEG
  (3)  026572  104404                            TRAP    C$BSEG
8098   026574  004737  004044            1$:     JSR     PC,CLRC                 ;CLEAR C BIT!
8099   026600  042737  000017  026616            BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
8100   026606  050537  026616                    BIS     R5,2$                   ;ADD ADDRESS TO INSTRUCTION
8101   026612                                    ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  026612  004537  003230                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8102   026616  010000                    2$:     010000                          ;LOAD MAR
8103   026620  042737  000017  026636            BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
8104   026626  050537  026636                    BIS     R5,3$                   ;ADD ADDRESS TO INSTRUCTION
8105   026632                                    ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  026632  004537  003230                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8106   026636  040740                    3$:     040400!<16*20>                  ;BR _ SUB
8107   026640                                    ROMCLK
  (1)  026640  004537  003230                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8108   026644  061224                            61224                           ;MOVE BR TO PORT4
8109   026646  111237  002636                    MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
8110   026652  116104  000004                    MOVB    4(R1),R4                ;PUT "FOUND" IN R4
8111   026656  123737  002636  002636            CMPB    $GDDAT,$GDDAT           ;DATA CORRECT?
8112   026664  001411                            BEQ     4$                      ;BR IF YES
8113   026666                                    ERROR   15,YES                  ;ALU ERROR
  (5)  026700  104455                            TRAP    C$ERDF
  (6)  026702  000017                            .WORD   15
  (6)  026704  004740                            .WORD   EM15
  (6)  026706  007104                            .WORD   ERR15
8114   026710                            4$:     ESCAPE  SEG
  (3)  026710  104410                            TRAP    C$ESCAPE
  (3)  026712  000014                            .WORD   10000$-.
8115   026714  005202                            INC     R2                      ;NEXT DATA
8116   026716  005205                            INC     R5                      ;NEXT ADDRESS
8117   026720  022705  000010                    CMP     #10,R5                  ;DONE YET?
8118   026724  001323                            BNE     1$                      ;BR IF NO
8119   026726                                    ENDSEG
  (3)  026726                            10000$:
  (3)  026726  104405                            TRAP    C$ESEG
8120   026730                                    EXIT    TST
  (3)  026730  104432                            TRAP    C$EXIT
  (3)  026732  000012                            .WORD   L10137-.
8121   026734     000     001     377     5$:     .BYTE   0,1,-1,0,0,253,125,0
       026737     000     000     253
       026742     125     000
8122
8123
8124                                      .EVEN
8125   026744                             ENDTST
  (3)  026744                             L10137:
  (3)  026744  104401                            TRAP    C$ETST
8126
8127
```

```
 8128  026746                                          BADHEAD
  (2)                                                ;******************** TEST 54 ********************
 8129                                                ;*ALU TEST
 8130                                                ;*TEST OF ALU FUNCTION ADD W/C WITH C BIT CLEARED
 8131                                                ;*ALU FUNCTION (A PLUS B PLUS C)        CODE 01
 8132                                                ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 8133                                                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 8134  026746                                          BADHEAD
  (2)                                                ;******************** TEST 54 ********************
 8135
 8136  026746                                BGNTST
  (3)  026746                                T54::
 8137  026746                                          MYINT
  (1)  026746  013701  002716                        MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 8138  026752                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  026752  004537  003142                        JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 8139  026756  005005                                CLR     R5              ;MEM + SP ADDRESS
 8140  026760  012702  027140                        MOV     #5$,R2          ;POINTER TO CORRECT DATA
 8141  026764  004737  003624                        JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 8142  026770  002654                                MEMDAT                  ;POINTER TO DATA
 8143  026772  004737  003776                        JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 8144  026776  002664                                SPDAT                   ;POINTER TO DATA
 8145  027000                                          BGNSEG
  (3)  027000  104404                                TRAP    C$BSEG
 8146  027002  004737  004044              1$:       JSR     PC,CLRC         ;CLEAR C BIT!
 8147  027006  042737  000017  027024                BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 8148  027014  050537  027024                        BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 8149  027020                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  027020  004537  003230                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8150  027024  010000                        2$:       010000                  ;LOAD MAR
 8151  027026  042737  000017  027044                BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 8152  027034  050537  027044                        BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 8153  027040                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  027040  004537  003230                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8154  027044  040420                        3$:       040400!<01*20>          ;BR   ADD W/C
 8155  027046                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  027046  004537  003230                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8156  027052  061224                                61224                   ;MOVE BR TO PORT4
 8157  027054  111237  002636                        MOVB    (R2),$GDDAT             ;PUT 'EXPECTED'' IN $GDDAT
 8158  027060  116104  000004                        MOVB    4(R1),R4        ;PUT 'FOUND'' IN R4
 8159  027064  123704  002636                        CMPB    $GDDAT,R4               ;DATA CORRECT?
 8160  027070  001411                                BEQ     4$              ;BR IS YES
 8161  027072                                          ERROR   15,YES          ;ALU ERROR
  (5)  027104  104455                                TRAP    C$ERDF
  (6)  027106  000017                                .WORD   15
  (6)  027110  004740                                .WORD   EM15
  (6)  027112  007104                                .WORD   ERR15
 8162  027114                                4$:       ESCAPE  SEG
  (3)  027114  104410                                TRAP    C$ESCAPE
  (3)  027116  000014                                .WORD   10000$-.
 8163  027120  005202                                INC     R2              ;NEXT DATA
 8164  027122  005205                                INC     R5              ;NEXT ADDRESS
 8165  027124  022705  000010                        CMP     #10,R5          ;DONE YET?
 8166  027130  001324                                BNE     1$              ;BR IF NO
 8167  027132                                          ENDSEG
  (3)  027132                                10000$:
```

```
  (3)  027132  104405                          TRAP    C$ESEG
8168   027134                                  EXIT    TST
  (3)  027134  104432                          TRAP    C$EXIT
  (3)  027136  000012                          .WORD   L10140-.
8169   027140     000     377     377  5$:     .BYTE   0,-1,-1,376,252,-1,-1,124
       027143     376     252     377
       027146     377     124
8170
8171                                  .EVEN
8172   027150                         ENDTST
  (3)  027150                         L10140:
  (3)  027150  104401                          TRAP    C$ETST
8173
8174
8175   027152                                 BADHEAD
  (2)                                          ;********************* TEST 55 ********************
8176                                           ;*ALU TEST
8177                                           ;*TEST OF ALU FUNCTION SUB W/C WITH C BIT CLEARED
8178                                           ;*ALU FUNCTION (A-B-C)  CODE=2
8179                                           ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8180                                           ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
8181   027152                                 BADHEAD
  (2)                                          ;********************* TEST 55 ********************
8182
8183   027152                         BGNTST
  (3)  027152                         T55::
8184   027152                                 MYINT
  (1)  027152  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
```

```
8186  027156                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  027156  004537  003142                         JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
8187  027162  005005                                 CLR     R5              ;MEM + SP ADDRESS
8188  027164  012702  027344                         MOV     #5$,R2          ;POINTER TO CORRECT DATA
8189  027170  004737  003624                         JSR     PC, MEMLD       ;LOAD 8 WORDS OF MAIN MEMORY
8190  027174  002654                                 MEMDAT                  ;POINTER TO DATA
8191  027176  004737  003776                         JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
8192  027202  002664                                 SPDAT                   ;POINTER TO DATA
8193  027204                                          BGNSEG
 (3)  027204  104404                                 TRAP    C$BSEG
8194  027206  004737  004044          1$:           JSR     PC,CLRC         ;CLEAR C BIT!
8195  027212  042737  000017  027230                 BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
8196  027220  050537  027230                         BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
8197  027224                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  027224  004537  003230                         JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8198  027230  010000                    2$:          010000                  ;LOAD MAR
8199  027232  042737  000017  027250                 BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
8200  027240  050537  027250                         BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
8201  027244                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  027244  004537  003230                         JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8202  027250  040440                    3$:          040400!<2*20>           ;BR _ SUB W/C
8203  027252                                          ROMCLK
 (1)  027252  004537  003230                         JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8204  027256  061224                                 61224                   ;MOVE BR TO PORT4
8205  027260  111237  002636                         MOVB    (R2),$GDDAT             ;PUT 'EXPECTED' IN $GDDAT
8206  027264  116104  000004                         MOVB    4(R1),R4        ;PUT 'FOUND' IN R4
8207  027270  123704  002636                         CMPB    $GDDAT,R4               ;DATA CORRECT?
8208  027274  001411                                 BEQ     4$              ;BR IF YES
8209  027276                                          ERROR   15,YES          ;ALU ERROR
 (5)  027310  104455                                 TRAP    C$ERDF
 (6)  027312  000017                                 .WORD   15
 (6)  027314  004740                                 .WORD   EM15
 (6)  027316  007104                                 .WORD   ERR15
8210  027320                    4$:                   FSCAPE  SEG
 (3)  027320  104410                                 TRAP    C$ESCAPE
 (3)  027322  000014                                 .WORD   10000$-.
8211  027324  005202                                 INC     R2                      ;NEXT DATA
8212  027326  005205                                 INC     R5                      ;NEXT ADDRESS
8213  027330  022705  000010                         CMP     #10,R5                  ;DONE YET?
8214  027334  001324                                 BNE     1$                      ;BR IF NO
8215  027336                                          ENDSEG
 (3)  027336                    10000$:
 (3)  027336  104405                                 TRAP    C$ESEG
8216  027340                                          EXIT    TST
 (3)  027340  104432                                 TRAP    C$EXIT
 (3)  027342  000012                                 .WORD   L10141-.
8217  027344     377     000     376  5$:           .BYTE   -1,0,376,-1,-1,252,124,-1
      027347     377     377     252
      027352     124     377
8218
8219
8220                                                .EVEN
8221  027354                                        ENDTST
 (3)  027354                                        L10141:
 (3)  027354  104401                                TRAP    C$ETST
8222
```

```
 8223
 8224  027356                                  BADHEAD
  (2)                                          ;********************* TEST 56 *********************
 8225                                          ;*ALU TEST
 8226                                          ;*TEST OF ALU FUNCTION INC A WITH C BIT CLEARED
 8227                                          ;*ALU FUNCTION (A PLUS 1)       CODE=3
 8228                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 8229                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 8230  027356                                  BADHEAD
  (2)                                          ;********************* TEST 56 *********************
 8231
 8232  027356                        BGNTST
  (3)  027356                        T56::
 8233  027356                                  MVINT
  (1)  027356  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 8234  027362                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  027362  004537  003142                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 8235  027366  012702  027550                MOV     #5$,R2          ;POINTER TO CORRECT DATA
 8236  027372  005005                         CLR     R5
 8237  027374  004737  003624                JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMRY
 8238  027400  002654                         MEMDAT                  ;POINTER TO DATA
 8239  027402  004737  003776                JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 8240  027406  002664                         SPDAT                   ;POINTER TO DATA
 8241  027410                                  BGNSEG
  (3)  027410  104404                        TRAP    C$BSEG
 8242  027412  004737  004044        1$:     JSR     PC,CLRC         ;CLEAR C BIT!
 8243  027416  042737  000017  027434        BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 8244  027424  050537  027434                BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 8245  027430                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  027430  004537  003230                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 8246  027434  010000                 2$:     010000                  ;LOAD MAR
 8247  027436  042737  000017  027454        BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 8248  027444  050537  027454                BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 8249  027450                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  027450  004537  003230                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 8250  027454  040460                 3$:     040400!<3*20>           ;BR  INC A
 8251  027456                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  027456  004537  003230                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 8252  027462  061224                         61224                   ;MOVE BR TO PORT4
 8253  027464  111237  002636                MOVB    (R2),$GDDAT             ;PUT 'EXPECTED' IN $GDDAT
 8254  027470  116104  000004                MOVB    4(R1),R4        ;PUT 'FOUND' IN R4
 8255  027474  123704  002636                CMPB    $GDDAT,R4               ;DATA CORRECT?
 8256  027500  001411                         BEQ     4$              ;BR IF YES
 8257  027502                                  ERROR   15,YES          ;ALU ERROR
  (5)  027514  104455                        TRAP    C$ERDF
  (6)  027516  000017                        .WORD   15
  (6)  027520  004740                        .WORD   EM15
  (6)  027522  007104                        .WORD   ERR15
 8258  027524                          4$:    ESCAPE  SEG
  (3)  027524  104410                        TRAP    C$ESCAPE
  (3)  027526  000014                        .WORD   10000$-.
 8259  027530  005202                         INC     R2              ;NEXT DATA
 8260  027532  005205                         INC     R5
 8261  027534  022705  000010                CMP     #10,R5          ;DONE YET?
 8262  027540  001324                         BNE     1$              ;BR IF NO
 8263  027542                                  ENDSEG
```

```
  (3)  027542                                  10000$:
  (3)  027542  104405                                  TRAP    C$ESEG
 8264  027544                                        EXIT    TST
  (3)  027544  104432                                  TRAP    C$EXIT
  (3)  027546  000012                                  .WORD   L10142-.
 8265  027550     001      001      000  5$:         .BYTE   1,1,0,0,126,126,253,253
       027553     000      126      126
       027556     253      253
 8266
 8267                                         .EVEN
 8268  027560                                 ENDTST
  (3)  027560                                 L10142:
  (3)  027560  104401                                  TRAP    C$ETST
 8269
 8270
 8271  027562                                        BADHEAD
  (2)                                                 ;********************* TEST 57 *********************
 8272                                                 ;*ALU TEST
 8273                                                 ;*TEST OF ALU FUNCTION 2A WITH C BIT CLEARED
 8274                                                 ;*ALU FUNCTION (A PLUS A)       CODE=5
 8275                                                 ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 8276                                                 ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 8277  027562                                        BADHEAD
  (2)                                                 ;********************* TEST 57 *********************
 8278
 8279  027562                                 BGNTST
  (3)  027562                                 T57::
 8280  027562                                        MYINT
  (1)  027562  013701  002716                          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 8281  027566                                        MSTCLR                  ;MASTER CLEAR DMC11
  (1)  027566  004537  003142                          JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 8282  027572  005005                                CLR     R5              ;MEM * SP ADDRESS
 8283  027574  012702  027754                        MOV     #5$,R2          ;POINTER TO CORRECT DATA
 8284  027600  004737  003624                        JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 8285  027604  002654                                MEMDAT                  ;POINTER TO DATA
 8286  027606  004737  003776                        JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 8287  027612  002664                                SPDAT                   ;POINTER TO DATA
 8288  027614                                        BGNSEG
  (3)  027614  104404                                  TRAP    C$BSEG
 8289  027616  004737  004044              1$:       JSR     PC,CLRC         ;CLEAR C BIT!
 8290  027622  042737  000017  027640                BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 8291  027630  050537  027640                        BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 8292  027634                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  027634  004537  003230                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 8293  027640  010000                        2$:       010000          ;LOAD MAR
 8294  027642  042737  000017  027660                BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 8295  027650  050537  027660                        BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 8296  027654                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  027654  004537  003230                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 8297  027660  040520                        3$:       040400!<5*20>           ;BR   2A
 8298  027662                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  027662  004537  003230                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 8299  027666  061224                                61224                   ;MOVE BR TO PORT4
 8300  027670  111237  002636                        MOVB    (R2),$GDDAT             ;PUT 'EXPECTED' IN $GDDAT
 8301  027674  116104  000004                        MOVB    4(R1),R4        ;PUT 'FOUND' IN R4
 8302  027700  123704  002636                        CMPB    $GDDAT,R4               ;DATA CORRECT?
```

```
 8303  027704  001411                          BEQ     4$              ;BR IF YES
 8304  027706                                  ERROR   15,YES          ;ALU ERROR
  (5)  027720  104455                          TRAP    C$ERDF
  (6)  027722  000017                          .WORD   15
  (6)  027724  004740                          .WORD   EM15
  (6)  027726  007104                          .WORD   ERR15
 8305  027730                          4$:     ESCAPE  SEG
  (3)  027730  104410                          TRAP    C$ESCAPE
  (3)  027732  000014                          .WORD   10000$-.
 8306  027734  005202                          INC     R2              ;NEXT DATA
 8307  027736  005205                          INC     R5              ;NEXT ADDRESS
 8308  027740  022705  000010                  CMP     #10,R5          ;DONE YET?
 8309  027744  001324                          BNE     1$              ;BR IF NO
 8310  027746                                  ENDSEG
  (3)  027746                          10000$:
  (3)  027746  104405                          TRAP    C$ESEG
 8311  027750                                  EXIT    TST
  (3)  027750  104432                          TRAP    C$EXIT
  (3)  027752  000012                          .WORD   L10143-.
 8312  027754     000      000      376  5$:     .BYTE   0,0,376,376,252,252,124,124
       027757     376      252      252
       027762     124      124
 8313
 8314                                          .EVEN
 8315  027764                                  ENDTST
  (3)  027764                          L10143:
  (3)  027764  104401                          TRAP    C$ETST
 8316
 8317
 8318  027766                                  BADHEAD
  (2)                                         ;******************** TEST 58 *********************
 8319                                         ;*ALU TEST
 8320                                         ;*TEST OF ALU FUNCTION A PLUS C WITH C BIT CLEARED
 8321                                         ;*ALU FUNCTION (A PLUS C)       CODE=4
 8322                                         ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 8323                                         ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 8324  027766                                  BADHEAD
  (2)                                         ;******************** TEST 58 *********************
 8325
 8326  027766                          BGNTST
  (3)  027766                          T58::
 8327  027766                                  MYINT
  (1)  027766  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 8328  027772                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  027772  004537  003142                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 8329  027776  005005                          CLR     R5              ;MEM + SP ADDRESS
 8330  030000  012702  030160                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
 8331  030004  004737  003624                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 8332  030010  002654                          MEMDAT                  ;POINTER TO DATA
 8333  030012  004737  003776                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 8334  030016  002664                          SPDAT                   ;POINTER TO DATA
 8335  030020                                  BGNSEG
  (3)  030020  104404                          TRAP    C$BSEG
 8336  030022  004737  004044          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
 8337  030026  042737  000017  030044          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 8338  030034  050537  030044                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
```

```
8339  030040
 (1)  030040  004537  003230            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
                                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8340  030044  010000              2$:   010000                  ;LOAD MAR
8341  030046  042737  000017  030064   BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
8342  030054  050537  030064           BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
8343  030060                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=304
 (1)  030060  004537  003230            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8344  030064  040500              3$:   040400!<4*20>           ;BR   A PLUS C
8345  030066                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCKL PC-5305
 (1)  030066  004537  003230            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8346  030072  061224                   61224                   ;MOVE BR TO PORT4
8347  030074  111237  002636           MOVB    (R2),$GDDAT             ;PUT ''EXPECTED'' IN $GDDAT
8348  030100  116104  000004           MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
8349  030104  123704  002636           CMPB    $GDDAT,R4               ;DATA CORRECT?
8350  030110  001411                   BEQ     4$              ;BR IS YES
8351  030112                           ERROR   15,YES          ;ALU ERROR
 (5)  030124  104455                   TRAP    C$ERDF
 (6)  030126  000017                   .WORD   15
 (6)  030130  004740                   .WORD   EM15
 (6)  030132  007104                   .WORD   ERR15
8352  030134              4$:   ESCAPE  SEG
 (3)  030134  104410                   TRAP    C$ESCAPE
 (3)  030136  000014                   .WORD   10000$-.
8353  030140  005202                   INC     R2              ;NEXT DATA
8354  030142  005205                   INC     R5              ;NEXT ADDRESS
8355  030144  022705  000010           CMP     #10,R5          ;DONE YET?
8356  030150  001324                   BNE     1$              ;BR IF NO
8357  030152                           ENDSEG
 (3)  030152                     10000$:
 (3)  030152  104405                   TRAP    C$ESEG
8358  030154                           EXIT    TST
 (3)  030154  104432                   TRAP    C$EXIT
 (3)  030156  000012                   .WORD   L10144-.
8359  030160     000     000     377   5$:   .BYTE   0,0,-1,-1,125,125,252,252
      030163     377     125     125
      030166     252     252
8360
8361                             .EVEN
8362  030170                     ENDTST
 (3)  030170                     L10144:
 (3)  030170  104401                   TRAP    C$ETST
8363
8364
8365  030172                           BADHEAD
 (2)                                   ;********** TEST 59 **********
8366                                   ;*ALU TEST
8367                                   ;*TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT CLEARED
8368                                   ;*ALU FUNCTION (A-B-1)  CODE=17
8369                                   ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8370                                   ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
8371  030172                           BADHEAD
 (2)                                   ;********** TEST 59 **********
8372
8373  030172                     BGNTST
 (3)  030172                     T59::
8374  030172                           MYINT
```

```
  (1)  030172  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
8375   030176                                MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  030176  004537  003142                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
8376   030202  005005                        CLR     R5              ;MEM + SP ADDRESS
8377   030204  012702  030364                MOV     #5$,R2          ;POINTER TO CORRECT DATA
8378   030210  004737  003624                JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
8379   030214  002654                        MEMDAT                  ;POINTER TO DATA
8380   030216  004737  003776                JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
8381   030222  002664                        SPDAT                   ;POINTER TO DATA
8382   030224                                BGNSEG
  (3)  030224  104404                        TRAP    C$BSEG
8383   030226  004737  004044          1$:   JSR     PC,CLRC         ;CLEAR C BIT!
8384   030232  042737  000017  030250        BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
8385   030240  050537  030250                BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
8386   030244                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  030244  004537  003230                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
8387   030250  010000                  2$:   010000                  ;LOAD MAR
8388   030252  042737  000017  030270        BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
8389   030260  050537  030270                BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
8390   030264                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  030264  004537  003230                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
8391   030270  040760                  3$:   040400!<17*20>          ;BR   2'S COMP SUB
8392   030272                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  030272  004537  003230                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
8393   030276  061224                        61224                   ;MOVE BR TO PORT4
8394   030300  111237  002636                MOVB    (R2),$GDDAT             ;PUT ''EXPECTED'' IN $GDDAT
8395   030304  116104  000004                MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
8396   030310  123704  002636                CMPB    $GDDAT,R4               ;DATA CORRECT?
8397   030314  001411                        BEQ     4$              ;BR IS YES
8398   030316                                ERROR   15,YES          ;ALU ERROR
  (5)  030330  104455                        TRAP    C$ERDF
  (6)  030332  000017                        .WORD   15
  (6)  030334  004740                        .WORD   EM15
  (6)  030336  007104                        .WORD   ERR15
8399   030340                          4$:   ESCAPE  SEG
  (3)  030340  104410                        TRAP    C$ESCAPE
  (3)  030342  000014                        .WORD   10000$-.
8400   030344  005202                        INC     R2              ;NEXT DATA
8401   030346  005205                        INC     R5              ;NEXT ADDRESS
8402   030350  022705  000010                CMP     #10,R5          ;DONE YET?
8403   030354  001324                        BNE     1$              ;BR IF NO
8404   030356                                ENDSEG
  (3)  030356                          10000$:
  (3)  030356  104405                        TRAP    C$ESEG
8405   030360                                EXIT    TST
  (3)  030360  104432                        TRAP    C$EXIT
  (3)  030362  000012                        .WORD   L10145-.
8406   030364     377     000     376  5$:   .BYTE   -1,0,376,-1,-1,252,124,-1
       030367     377     377     252
       030372     124     377
8407
8408                                  .EVEN
8409   030374                         ENDTST
  (3)  030374                         L10145:
  (3)  030374  104401                        TRAP    C$ETST
8410
```

```
 8411
 8412  030376                                  BADHEAD
  (2)                                          ;******************** TEST 60 ********************
 8413                                          ;*ALU TEST
 8414                                          ;*TEST OF ALU FUNCTION DEC A WITH C BIT CLEARED
 8415                                          ;*ALU FUNCTION (A-1)    CODE=7
 8416                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 8417                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 8418
 8419  030376                                  BADHEAD
  (2)                                          ;******************** TEST 60 ********************
 8420
 8421  030376                        BGNTST
  (3)  030376                        T60::
 8422  030376                                  MYINT
  (1)  030376  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 8423  030402                                  MSTCLR                  ;MASTER CLEAR DMC11
  (1)  030402  004537  003142                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 8424  030406  005005                        CLR     R5              ;MEM + SP ADDRESS
 8425  030410  012702  030570                MOV     #5$,R2          ;POINTER TO CORRECT DATA
 8426  030414  004737  003624                JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMMOR
 8427  030420  002654                        MEMDAT                  ;POINTER TO DATA
 8428  030422  004737  003776                JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 8429  030426  002664                        SPDAT                   ;POINTER TO DATA
 8430  030430                                  BGNSEG
  (3)  030430  104404                        TRAP    C$BSEG
 8431  030432  004737  004044        1$:     JSR     PC,CLRC         ;CLEAR C BIT!
 8432  030436  042737  000017  030454        BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 8433  030444  050537  030454                BIS     R5,2$           ;ADD ADDRESS TO INSTRUCITON
 8434  030450                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  030450  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8435  030454  010000                2$:     010000                  ;LOAD MAR
 8436  030456  042737  000017  030474        BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 8437  030464  050537  030474    .           BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 8438  030470                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  030470  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8439  030474  040560                3$:     040400!<7*20>           ;BR   DEC A
 8440  030476                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  030476  004537  003230                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8441  030502  061224                        61224                   ;MOVE BR TO PORT4
 8442  030504  111237  002636                MOVB    (R2),$GDDAT             ;PUT 'EXPECTED'' IN $GDDAT
 8443  030510  116104  000004                MOVB    4(R1),R4        ;PUT 'FOUND'' IN R4
 8444  030514  123704  002636                CMPB    $GDDAT,R4               ;DATA CORRECT?
 8445  030520  001411                        BEQ     4$              ;BR IF YES
 8446  030522                                  ERROR   15,YES          ;ALU ERROR
  (5)  030534  104455                        TRAP    C$ERDF
  (6)  030536  000017                        .WORD   15
  (6)  030540  004740                        .WORD   EM15
  (6)  030542  007104                        .WORD   ERR15
 8447  030544                        4$:     ESCAPE  SEG
  (3)  030544  104410                        TRAP    C$ESCAPE
  (3)  030546  000014                        .WORD   10000$-.
 8448  030550  005202                        INC     R2              ;NEXT DATA
 8449  030552  005205                        INC      R5             ;NEXT ADDRESS
 8450  030554  022705  000010                CMP     #10,R5          ;DONE YET?
 8451  030560  001324                        BNE      1$             ;BR IF NO
```

```
 8452  030562                                          ENDSEG
  (3)  030562                                  10000$:
  (3)  030562  104405                                 TRAP    C$ESEG
 8453  030564                                          EXIT    TST
  (3)  030564  104432                                 TRAP    C$EXIT
  (3)  030566  000012                                 .WORD   L10146-.
 8454  030570     377     377     376  5$:     .BYTE   -1,-1,376,376,124,124,251,251
       030573     376     124     124
       030576     251     251
 8455
 8456                                         .EVEN
 8457  030600                                 ENDTST
  (3)  030600                                 L10146:
  (3)  030600  104401                                 TRAP    C$ETST
 8458
 8459
 8460  030602                                          BADHEAD
  (2)                                                ;****************** TEST 61 ******************
 8461                                                ;*ALU TEST
 8462                                                ;*TEST OF ALU FUNCTION SEL B WITH C BIT SET
 8463                                                ;*ALU FUNCTION (B)      CODE=11
 8464                                                ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 8465                                                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 8466  030602                                          BADHEAD
  (2)                                                ;****************** TEST 61 ******************
 8467
 8468  030602                                 BGNTST
  (3)  030602                                 T61::
 8469  030602                                          MYINT
  (1)  030602  013701  002716                         MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 8470  030606                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  030606  004537  003142                         JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 8471  030612  005005                                 CLR     R5              ;MEM + SP ADDRESS
 8472  030614  012702  030774                         MOV     #5$,R2          ;POINTER TO CORRECT DATA
 8473  030620  004737  003624                         JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 8474  030624  002654                                 MEMDAT                  ;POINTER TO DATA
 8475  030626  004737  003776                         JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 8476  030632  002664                                 SPDAT                   ;POINTET TO DATA
 8477  030634                                          BGNSEG
  (3)  030634  104404                                 TRAP    C$BSEG
 8478  030636  004737  004062              1$:       JSR     PC,SETC         ;SET C BIT!
 8479  030642  042737  000017  030660                 BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 8480  030650  050537  030660                         BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 8481  030654                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
  (1)  030654  004537  003230                         JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8482  030660  010000                       2$:       010000          ;LOAD MAR
 8483  030662  042737  000017  030700                 BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 8484  030670  050537  030700                         BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 8485  030674                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  030674  004537  003230                         JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8486  030700  040620                       3$:       040400.<11*20>  ;BR   SEL B
 8487  030702                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  030702  004537  003230                         JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8488  030706  061224                                 61224           ;MOVE BR TO PORT4
 8489  030710  111237  002636                         MOVB    (R2),$GDDAT             ;PUT ''EXPECTED'' IN $GDDAT
 8490  030714  116104  000004                         MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
```

```
 8491  030720  123704  002636                 CMPB    $GDDAT,R4               ;DATA CORRECT?
 8492  030724  001411                         BEQ     4$                      ;BR IF YES
 8493  030726                                 ERROR   23,YES                  ;ALU ERROR
  (5)  030740  104455                         TRAP    C$ERDF
  (6)  030742  000027                         .WORD   23
  (6)  030744  005212                         .WORD   EM23
  (6)  030746  007370                         .WORD   ERR23
 8494  030750                          4$:    ESCAPE  SEG
  (3)  030750  104410                         TRAP    C$ESCAPE
  (3)  030752  000014                         .WORD   10000$-.
 8495  030754  005202                         INC     R2                      ;NEXT DATA
 8496  030756  005205                         INC     R5                      ;NEXT ADDRESS
 8497  030760  022705  000010                 CMP     #10,R5                  ;DONE YET?
 8498  030764  001324                         BNE     1$                      ;BR IF NO
 8499  030766                                 ENDSEG
  (3)  030766                          10000$:
  (3)  030766  104405                         TRAP    C$ESEG
 8500  030770                                 EXIT    TST
  (3)  030770  104432                         TRAP    C$EXIT
  (3)  030772  000012                         .WORD   L10147-.
 8501  030774     000     377     000  5$:    .BYTE   0,-1,0,-1,125,252,125,252
       030777     377     125     252
       031002     125     252
 8502
 8503                                 .EVEN
 8504  031004                         ENDTST
  (3)  031004                         L10147:
  (3)  031004  104401                         TRAP    C$ETST
 8505
 8506
 8507  031006                                 BADHEAD
  (2)                                        ;******************** TEST 62 **********************
 8508                                        ;*ALU TEST
 8509                                        ;*TEST OF ALU FUNCTION SEL A WITH C BIT SET
 8510                                        ;*ALU FUNCTION (A)      CODE=10
 8511                                        ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 8512                                        ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 8513  031006                                 BADHEAD
  (2)                                        ;******************** TEST 62 **********************
```

```
8515
8516
8517  031006                          BGNTST
 (3)  031006                          T62::
8518  031006                                  MYINT
 (1)  031006  013701  002716                   MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
8519  031012                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  031012  004537  003142                   JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
8520  031016  005005                          CLR     R5              ;MEM + SP ADDRESS
8521  031020  012702  031200                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
8522  031024  004737  003624                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
8523  031030  002654                          MEMDAT                  ;POINTER TO DATA
8524  031032  004737  003776                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
8525  031036  002664                          SPDAT                   ;POINTER TO DATA
8526  031040                                  BGNSEG
 (3)  031040  104404                          TRAP    C$BSEG
8527  031042  004737  004062          1$:     JSR     PC,SETC         ;SET C BIT!
8528  031046  042737  000017  031064          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
8529  031054  050537  031064                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
8530  031060                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
 (1)  031060  004537  003230                   JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8531  031064  010000                  2$:     010000                  ;LOAD MAR
8532  031066  042737  000017  031104          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCITON
8533  031074  050537  031104                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
8534  031100                                  ROMCLK                  ;NEXT WORD IS INSTRUCITON, ROMCLK PC=5304
 (1)  031100  004537  003230                   JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8535  031104  040600                  3$:     040400.<10*20>          ;BR   SEL A
8536  031106                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  031106  004537  003230                   JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8537  031112  061224                          61224                   ;MOVE BR TO PORT4
8538  031114  111237  002636                  MOVB    (R2),$GDDAT             ;PUT 'EXPECTED'' IN $GDDAT
8539  031120  116104  000004                  MOVB    4(R1),R4        ;PUT 'FOUND'' IN R4
8540  031124  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
8541  031130  001411                          BEQ     4$              ;BR IF YES
8542  031132                                  ERROR   23,YES          ;ALU ERROR
 (5)  031144  104455                          TRAP    C$ERDF
 (6)  031146  000027                          .WORD   23
 (6)  031150  005212                          .WORD   EM23
 (6)  031152  007370                          .WORD   ERR23
8543  031154                          4$:     ESCAPE  SEG
 (3)  031154  104410                          TRAP    C$ESCAPE
 (3)  031156  000C14                          .WORD   10000$-.
8544  031160  005202                          INC     R2              ;NEXT DATA
8545  031162  005205                          INC     R5              ;NEXT ADDRESS
8546  031164  022705  000010                  CMP     #10,R5          ;DONE YET?
8547  031170  001324                          BNE     1$              ;BR IF NO
8548  031172                                  ENDSEG
 (3)  031172                          10000$:
 (3)  031172  104405                          TRAP    C$ESEG
8549  031174                                  EXIT    TST
 (3)  031174  104432                          TRAP    C$EXIT
 (3)  031176  000012                          .WORD   L10150-.
8550  031200     000      000      377  5$:     .BYTE   0,0,-1,-1,125,125,252,252
      031203     377      125      125
      031206     252      252
8551
```

```
8552
8553  031210                                  .EVEN
8553  031210                                  ENDTST
 (3)  031210                                  L10150:
 (3)  031210  104401                                 TRAP    C$ETST
8554
8555
8556  031212                                        BADHEAD
 (2)                                                ;********** TEST 63 **********
8557                                                ;*ALU TEST
8558                                                ;*TEST OF ALU FUNCTION A OR NOTB WITH C BIT SET
8559                                                ;*ALU FUNCTION (A OR NOTB)      CODE=12
8560                                                ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8561                                                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
8562  031212                                        BADHEAD
 (2)                                                ;********** TEST 63 **********
8563
8564  031212                                  BGNTST
 (3)  031212                                  T63::
8565  031212                                        MYINT
 (1)  031212  013701  002716                        MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
8566  031216                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  031216  004537  003142                        JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
8567  031222  005005                               CLR     R5              ;MEM + SP ADDRESS
8568  031224  012702  031404                       MOV     #5$,R2          ;POINTER TO CORRECT DATA
8569  031230  004737  003624                       JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
8570  031234  002654                               MEMDAT                  ;POINTER TO DATA
8571  031236  004737  003776                       JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
8572  031242  002664                               SPDAT                   ;POINTER TO DATA
8573  031244                                        BGNSEG
 (3)  031244  104404                                TRAP    C$BSEG
8574  031246  004737  004062          1$:          JSR     PC,SETC         ;SET C BIT!
8575  031252  042737  000017  031270               BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
8576  031260  050537  031270                       BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
8577  031264                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  031264  004537  003230                        JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
8578  031270  010000                       2$:      010000                  ;LOAD MAR
8579  031272  042737  000017  031310               BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
8580  031300  050537  031310                       BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
8581  031304                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  031304  004537  003230                        JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
8582  031310  040640                       3$:      040400!<12*20>          ;BR   A OR NOTB
8583  031312                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  031312  004537  003230                        JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
8584  031316  061224                               61224                   ;MOVE BR TO PORT4
8585  031320  111237  002636                       MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
8586  031324  116104  000004                       MOVB    4(R1),R4        ;PUT "FOUND" IN R4
8587  031330  123704  002636                       CMPB    $GDDAT,R4               ;DATA CORRECT?
8588  031334  001411                               BEQ     4$              ;BR IF YES
8589  031336                                        ERROR    15,YES         ;ALU ERROR
 (5)  031350  104455                                TRAP    C$ERDF
 (6)  031352  000017                               .WORD   15
 (6)  031354  004740                               .WORD   EM15
 (6)  031356  007104                               .WORD   ERR15
8590  031360                                4$:     ESCAPE  SEG
 (3)  031360  104410                                TRAP    C$ESCAPE
 (3)  031362  000014                               .WORD   10000$-.
```

```
 8591  031364  005202                          INC     R2              ;NEXT DATA
 8592  031366  005205                          INC     R5              ;NEXT ADDRESS
 8593  031370  022705  000010                  CMP     #10,R5          ;DONE YET?
 8594  031374  001324                          BNE     1$              ;BR IF NO
 8595  031376                                  ENDSEG
  (3)  031376                          10000$:
  (3)  031376  104405                          TRAP    C$ESEG
 8596  031400                                  EXIT    TST
  (3)  031400  104432                          TRAP    C$EXIT
  (3)  031402  000012                          .WORD   L10151-.
 8597  031404     377     000     377  5$:     .BYTE   -1,0,-1,-1,-1,125,252,-1
       031407     377     377     125
       031412     252     377
 8598
 8599                                 .EVEN
 8600  031414                         ENDTST
  (3)  031414                         L1C151:
  (3)  031414  104401                          TRAP    C$ETST
 8601
 8602
 8603  031416                                  BADHEAD
  (2)                                         ;********************* TEST 64 *********************
 8604                                         ;*ALU TEST
 8605                                         ;*TEST OF ALU FUNCTION A AND B WITH C BIT SET
 8606                                         ;ALU FUNCTION (A AND B) CODE=13
 8607                                         ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 8608                                         ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 8609  031416                                  BADHEAD
  (2)                                         ;********************* TEST 64 *********************
 8610
 8611  031416                         BGNTST
  (3)  031416                         T64::
 8612  031416                                  MYINT
  (1)  031416  013701  002716                 MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 8613  031422                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  031422  004537  003142                 JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 8614  031426  005005                          CLR     R5              ;MEM + SP ADDRESS
 8615  031430  012702  031610                 MOV     #5$,R2          ;POINTER TO CORRECT ADDRESS
 8616  031434  004737  003624                 JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 8617  031440  002654                          MEMDAT                  ;POINTER TO DATA
 8618  031442  004737  003776                 JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 8619  031446  002664                          SPDAT                   ;POINTER TO DATA
 8620  031450                                  BGNSEG
  (3)  031450  104404                          TRAP    C$BSEG
 8621  031452  004737  004062          1$:     JSR     PC,SETC         ;SET C BIT!
 8622  031456  042737  000017  031474         BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 8623  031464  050537  031474                 BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 8624  031470                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  031470  004537  003230                 JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8625  031474  010000                  2$:     010000                  ;LOAD MAR
 8626  031476  042737  000017  031514         BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 8627  031504  050537  031514                 BIS     R5,3$           ;ADD ADDRESS TO INSTRUCITON
 8628  031510                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  031510  004537  003230                 JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8629  031514  040660                  3$:     040400.<13*20>          ;BR   A AND B
 8630  031516                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
```

```
  (1)  031516  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8631  031522  061224                          61224                   ;MOVE BR TO PORT4
 8632  031524  111237  002636                  MOVB    (R2),$GDDAT             ;PUT 'EXPECTED''IN $GDDAT
 8633  031530  116104  000004                  MOVB    4(R1),R4        ;PUT 'FOUND'' IN R4
 8634  031534  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
 8635  031540  001411                          BEQ     4$              ;BR IF YES
 8636  031542                                  ERROR   23,YES          ;ALU ERROR
  (5)  031554  104455                          TRAP    C$ERDF
  (6)  031556  000027                          .WORD   23
  (6)  031560  005212                          .WORD   EM23
  (6)  031562  007370                          .WORD   ERR23
 8637  031564                          4$:     ESCAPE  SEG
  (3)  031564  104410                          TRAP    C$ESCAPE
  (3)  031566  000014                          .WORD   10000$-.
 8638  031570  005202                          INC     R2              ;NEXT DATA
 8639  031572  005205                          INC     R5              ;NEXT ADDRESS
 8640  031574  022705  000010                  CMP     #10,R5          ;DONE YET?
 8641  031600  001324                          BNE     1$              ;BR IF NO
 8642  031602                                  ENDSEG
  (3)  031602                          10000$:
  (3)  031602  104405                          TRAP    C$ESEG
 8643  031604                                  EXIT    TST
  (3)  031604  104432                          TRAP    C$EXIT
  (3)  031606  000012                          .WORD   L10152-.
 8644  031610     000      000      000  5$:     .BYTE   0,0,0,-1,125,0,0,252
       031613     377      125      000
       031616     000      252
 8645
 8646                                  .EVEN
 8647  031620                          ENDTST
  (3)  031620                          L10152:
  (3)  031620  104401                          TRAP    C$ETST
 8648
 8649
 8650  031622                                  BADHEAD
  (2)                                         ;********************* TEST 65 *********************
 8651                                         ;*ALU TEST
 8652                                         ;*TEST OF ALU FUNCTION A OR B WITH C BIT SET
 8653                                         ;*ALU FUNCTION (A OR B) CODE=14
 8654                                         ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 8655                                         ;*PERFORM T4E FUNCTION, VERIFY THE RESULTS
 8656  031622                                  BADHEAD
  (2)                                         ;********************* TEST 65 *********************
 8657
 8658  031622                          BGNTST
  (3)  031622                          T65::
 8659  031622                                  MYINT
  (1)  031622  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 8660  031626                                  MSTCLR                  ;MASTER C EAR M8200,4,7
  (1)  031626  004537  003142                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 8661  031632  005005                          CLR     R5              ;MEM + SP ADDRESS
 8662  031634  012702  032014                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
 8663  031640  004737  003624                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 8664  031644  002654                          MEMDAT                  ;POINTER TO DATA
 8665  031646  004737  003776                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 8666  031652  002664                          SPDAT                   ;POINTER TO DATA
```

```
 8667  031654
  (3)  031654  104404
 8668  031656  004737  004062          1$:     JSR     PC,SETC         ;SET C BIT!
 8669  031662  042737  000017  031700          BIC #17,2$              ;CLEAR ADDRESS FIELD OF INSTRUCTION
 8670  031670  050537  031700                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 8671  031674                                  ROMCLK                  ;NEXT WORD IS INSTRUCITON, ROMCLK PC-5304
  (1)  031674  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8672  031700  010000                  2$:     010000                  ;LOAD MAR
 8673  031702  042737  000017  031720          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 8674  031710  050537  031720                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 8675  031714                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  031714  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8676  031720  040700                  3$:     040400!<14*20>          ;BR   A OR B
 8677  031722                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  031722  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8678  031726  061224                          61224                   ;MOVE BR TO PORT4
 8679  031730  111237  002636                  MOVB    (R2),$GDDAT             ;PUT 'EXPECTED''IN R4
 8680  031734  116104  000004                  MOVB    4(R1),R4        ;PUT 'FOUND'' IN R4
 8681  031740  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
 8682  031744  001411                          BEQ     4$              ;BR IF YES
 8683  031746                                  ERROR   23,YES          ;ALL ERROR
  (5)  031760  104455                          TRAP    C$ERDF
  (6)  031762  000027                          .WORD   23
  (6)  031764  005212                          .WORD   EM23
  (6)  031766  007370                          .WORD   ERR23
 8684  031770                          4$:     ESCAPE  SEG
  (3)  031770  104410                          TRAP    C$ESCAPE
  (3)  031772  000014                          .WORD   10000$-.
 8685  031774  005202                          INC     R2                      ;NEXT DATA
 8686  031776  005205                          INC     R5                      ;NEXT ADDRESS
 8687  032000  022705  000010                  CMP     #10,R5                  ;DONE YET?
 8688  032004  001324                          BNE     1$                      ;BR IF NO
 8689  032006                                  ENDSEG
  (3)  032006                      10000$:
  (3)  032006  104405                          TRAP    C$ESEG
 8690  032010                                  EXIT    TST
  (3)  032010  104432                          TRAP    C$EXIT
  (3)  032012  000012                          .WORD   L10153-.
 8691  032014     000     377     377  5$:     .BYTE   0,-1,-1,-1,125,-1,-1,252
       032017     377     125     377
       032022     377     252
 8692
 8693                                     .EVEN
 8694  032024                             ENDTST
  (3)  032024                             L10153:
  (3)  032024  104401                          TRAP    C$ETST
 8695
 8696
 8697  032026                                  BADHEAD
  (2)                                         ;********** TEST 66 **********
 8698                                         ;*ALU TEST
 8699                                         ;*TEST OF ALU FUNCTION A XOR B WITH C BIT SET
 8700                                         .*ALU FUNCTION (A XOR B)         CODE=15
 8701                                         ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 8702                                                 ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 8703
```

```
 8704  032026                                          BADHEAD
  (2)
 8705  032026                                  BGNTST
  (3)  032026                                  T66::
 8706  032026                                          MYINT
  (1)  032026  013701  002716                         MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 8707  032032                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  032032  004537  003142                         JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 8708  032036  005005                                 CLR     R5              ;MEM + SP ADDRESS
 8709  032040  012702  032220                         MOV     #5$,R2          ;POINTER TO CORRECT DATA
 8710  032044  004737  003624                         JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
 8711  032050  002654                                 MEMDAT                  ;POINTER TO DATA
 8712  032052  004737  003776                         JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 8713  032056  0026.4                                 SPDAT                   ;POINTER TO DATA
 8714  032060                                          BGNSEG
  (3)  032060  104404                                 TRAP    C$BSEG
 8715  032062  004737  004062                 1$:     JSR     PC,SETC         ;SET C BIT!
 8716  032066  042737  000017  032104                 BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 8717  032074  050537  032104                         BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 8718  032100                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  032100  004537  003230                         JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8719  032104  010000                         2$:     010000                  ;LOAD MAR
 8720  032106  042737  000017  032124                 BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 8721  032114  050537  032124                         BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 8722  032120                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  032120  004537  003230                         JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8723  032124  040720                         3$:     040400!<15*20>          ;BR   A XOR B
 8724  032126                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  032126  004537  003230                         JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8725  032132  061224                                 61224                   ;MOVE BR TO PORT4
 8726  032134  111237  002636                         MOVB    (R2),$GDDAT             ;PUT 'EXPECTED" IN $GDDAT
 8727  032140  116104  000004                         MOVB    4(R1),R4        ;PUT 'FOUND" IN R4
 8728  032144  123704  002636                         CMPB    $GDDAT,R4               ;DATA CORRECT?
 8729  032150  001411                                 BEQ     4$              ;BR IF YES
 8730  032152                                          ERROR   23,YES          ;ALU ERROR
  (5)  032164  104455                                 TRAP    C$ERDF
  (6)  032166  000027                                 .WORD   23
  (6)  032170  005212                                 .WORD   EM23
  (6)  032172  007370                                 .WORD   ERR23
 8731  032174                                  4$:     ESCAPE  SEG
  (3)  032174  104410                                 TRAP    C$ESCAPE
  (3)  032176  000014                                 .WORD   10000$-.
 8732  032200  005202                                 INC     R2              ;NEXT DATA
 8733  032202  005205                                 INC     R5              ;NEXT ADDRESS
 8734  032204  022705  000010                         CMP     #10,R5          ;DONE YET?
 8735  032210  001324                                 BNE     1$              ;BR IF NO
 8736  032212                                          ENDSEG
  (3)  032212                                 10000$:
  (3)  032212  104405                                 TRAP    C$ESEG
 8737  032214                                          EXIT    TST
  (3)  032214  104432                                 TRAP    C$EXIT
  (3)  032216  000012                                 .WORD   L10154-.
 8738  032220     000     377     377          5$:     .BYTE   0,-1,-1,0.0,-1,-1,0
       032223     000     000     377
       032226     377     000
 8739
```

```
 8740                                   .EVEN
 8741  032230                           ENDTST
  (3)  032230                           L10154:
  (3)  032230  104401                           TRAP    C$ETST
 8742
 8743
 8744  032232                                  BADHEAD
  (2)                                          ;******************** TEST 67 ********************
 8745                                          ;*ALU TEST
 8746                                          ;*TEST OF ALU FUNCTION ADD WITH C BIT SET
 8747                                          ;*ALU FUNCTION (A PLUS B)       CODE=00
 8748                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 8749                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 8750  032232                                 -BADHEAD
  (2)                                          ;******************** TEST 67 ********************
 8751
 8752  032232                          BGNTST
  (3)  032232                          T67::
 8753  032232                                  MYINT
  (1)  032232  013701  002716                 MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 8754  032236                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  032236  004537  003142                 JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 8755  032242  005005                         CLR     R5              ;MEM + SP ADDRESS
 8756  032244  012702  032424                 MOV     #5$,R2          ;POINTER TO CORRECT DATA
 8757  032250  004737  003624                 JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 8758  032254  002654                         MEMDAT                  ;POINTER TO DATA
 8759  032256  004737  003776                 JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 8760  032262  002664                         SPDAT                   ;OOINTER TO DATA
 8761  032264                                  BGNSEG
  (3)  032264  104404                         TRAP    C$BSEG
 8762  032266  004737  004062          1$:     JSR     PC,SETC         ;SET C BIT!
 8763  032272  042737  000017  032310         BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 8764  032300  050537  032310                 BIS     R5,2$           ;ADD ADDRESS TO INSTRUCITON
 8765  032304                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  032304  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 8766  032310  010000                  2$:     010000                  ;LOAD MAR
 8767  032312  042737  000017  032330         BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 8768  032320  050537  032330                 BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 8769  032324                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
  (1)  032324  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 8770  032330  040400                  3$:     040400!<00*20>          ;BR   ADD
 8771  032332                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  032332  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 8772  032336  061224                         61224                   ;MOVE BR TO PORT4
 8773  032340  111237  002636                 MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
 8774  032344  116104  000004                 MOVB    -4(R1),R4       ;PUT "FOUND" IN R4
 8775  032350  123704  002636                 CMPB    $GDDAT,R4               ;DATA CORRECT?
 8776  032354  001411                         BEQ     4$              ;BR IF YES
 8777  032356                                  ERROR   23,YES          ;ALU ERROR
  (5)  032370  104455                         TRAP    C$ERDF
  (6)  032372  000027                         .WORD   23
  (6)  032374  005212                         .WORD   EM23
  (6)  032376  007370                         .WORD   ERR23
 8778  032400                           4$:    ESCAPE  SEG
  (3)  032400  104410                         TRAP    C$ESCAPE
  (3)  032402  000014                         .WORD   10000$-.
```

```
 8779  032404  005202                          INC     R2              ;NEXT DATA
 8780  032406  005205                          INC     R5              ;NEXT ADDRESS
 8781  032410  022705  000010                  CMP     #10,R5          ;DONE YET?
 8782  032414  001324                          BNE     1$              ;BR IF NO
 8783  032416                                  ENDSEG
  (3)  032416                          10000$:
  (3)  032416  104405                          TRAP    C$ESEG
 8784  032420                                  EXIT    TST
  (3)  032420  104432                          TRAP    C$EXIT
  (3)  032422  000012                          .WORD   L10155-.
 8785  032424     000     377     377  5$:     .BYTE   0,-1,-1,376,252,-1,-1,124
       032427     376     252     377
       032432     377     124
 8786
 8787                                 .EVEN
 8788  032434                         ENDTST
  (3)  032434                         L10155:
  (3)  032434  104401                          TRAP    C$ETST
 8789
 8790
 8791  032436                                  BADHEAD
  (2)                                         ;******************** TEST 68 ********************
 8792                                         ;*ALU TEST
 8793                                         ;*TEST OF ALU FUNCTION 2A W/C WITH C BIT SET
 8794                                         ;*ALU FUNCTION (A PLUS A PLUS C)        CODE=6
 8795                                         ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 8796                                         ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 8797  032436                                  BADHEAD
  (2)                                         ;******************** TEST 68 ********************
 8798
 8799  032436                         BGNTST
  (3)  032436                         T68::
 8800  032436                                  MYINT
  (1)  032436  013701  002716                 MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 8801  032442                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  032442  004537  003142                 JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 8802  032446  005005                          CLR     R5              ;MEM + SP ADDRESS
 8803  032450  012702  032630                 MOV     #5$,R2          ;POINTER TO CORRECT DATA
 8804  032454  004737  003624                 JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 8805  032460  002654                          MEMDAT                  ;POINTER TO DATA
 8806  032462  004737  003776                 JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 8807  032466  002664                          SPDAT                   ;POINTER TO DATA
 8808  032470                                  BGNSEG
  (3)  032470  104404                          TRAP    C$BSEG
 8809  032472  004737  004062          1$:     JSR     PC,SETC         ;SET C BIT!
 8810  032476  042737  000017  032514         BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 8811  032504  050537  032514                 BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 8812  032510                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  032510  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 8813  032514  010000                  2$:     010000                  ;LOAD MAR
 8814  032516  042737  000017  032534         BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 8815  032524  050537  032534                 BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 8816  032530                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-55304
  (1)  032530  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 8817  032534  040540                  3$:     040400.<6*20>           ;BR   2A W/C
 8818  032536                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
```

```
  (1)  032536  004537  003230                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8819   032542  061224                            61224                   ;MOVE BR TO PORT4
8820   032544  111237  002636                    MOVB    (R2),$GDDAT             ;PUT 'WXPECTED' IN $GDDAT
8821   032550  116104  000004                    MOVB    4(R1),R4        ;PUT 'FOUND' IN R4
8822   032554  123704  002636                    CMPB    $GDDAT,R4               ;DATA CORRECT?
8823   032560  001411                            BEQ     4$              ;BR IF YES
8824   032562                                    FRROR   23,YES          ;ALU ERROR
  (5)  032574  104455                            TRAP    C$ERDF
  (6)  032576  000027                            .WORD   23
  (6)  032600  005212                            .WORD   EM23
  (6)  032602  007370                            .WORD   ERR23
8825   032604                            4$:     ESCAPE  SEG
  (3)  032604  104410                            TRAP    C$ESCAPE
  (3)  032606  000014                            .WORD   10000$-.
8826   032610  005202                            INC     R2              ;NEXT DATA
8827   032612  005205                            INC     R5              ;NEXT ADDRESS
8828   032614  022705  000010                    CMP     #10,R5          ;DONE YET?
8829   032620  001324                            BNE     1$              ;BR IF NO
8830   032622                                    ENDSEG
  (3)  032622                            10000$:
  (3)  032622  104405                            TRAP    C$ESEG
8831   032624                                    EXIT    TST
  (3)  032624  104432                            TRAP    C$EXIT
  (3)  032626  000012                            .WORD   L10156-.
8832   032630     001     001     377  5$:      .BYTE   1,1,-1,-1,253,253,125,125
       032633     377     253     253
       032636     125     125
8833
8834                                      .EVEN
8835   032640                             ENDTST
  (3)  032640                             L10156:
  (3)  032640  104401                             TRAP    C$ETST
8836
8837
8838   032642                                    BADHEAD
  (2)                                             ;********** TEST 69 **********
8839                                              ;*ALU TEST
8840                                              ;*TEST OF ALU FUNCTION SUB WITH C BIT SET
8841                                              ;*ALU FUNCTION (A-B)    CODE=16
8842                                              ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8843                                              ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
8844   032642                                     BADHEAD
  (2)                                             ;********** TEST 69 **********
8845
8846   032642                             BGNTST
  (3)  032642                             T69::
8847   032642                                     MYINT
  (1)  032642  013701  002716                     MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
8848   032646                                     MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  032646  004537  003142                     JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
8849   032652  005005                             CLR     R5              ;MEM + SP ADDRESS
8850   032654  012702  033034                     MOV #5$,R2              ;POINTER TO CORRECT DATA
8851   032660  004737  003624                     JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
8852   032664  002654                             MEMDAT                  ;POINTER TO DATA
8853   032666  004737  003776                     JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
8854   032672  002664                             SPDAT                   ;POINTER TO DATA
```

```
8855  032674
 (3)  032674  104404
8856  032676  004737  004062          1$:   JSR     PC,SETC         ;SET C BIT!
8857  032702  042737  000017  032720        BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
8858  032710  050537  032720              BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
8859  032714                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  032714  004537  003230              JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
8860  032720  010000                2$:   010000                  ;LOAD MAR
8861  032722  042737  000017  032740        BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
8862  032730  050537  032740              BIS     R5,3$           ;ADD ADDRESS  TO INSTRUCTION
8863  032734                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  032734  004537  003230              JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
8864  032740  040740                3$:   040400!<16*20>          ;BR   SUB
8865  032742                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  032742  004537  003230              JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
8866  032746  061224                      61224                   ;MOVE BR TO PORT4
8867  032750  111237  002636              MOVB    (R2),$GDDAT             ;PUT 'EXPECTED' IN $GDDAT
8868  032754  116104  000004              MOVB    4(R1),R4        ;PUT 'FOUND' IN R4
8869  032760  123704  002636              CMPB    $GDDAT,R4               ;DATA CORRECT?
8870  032764  001411                      BEQ     4$              ;BR IF YES
8871  032766                              ERROR   23,YES          ;ALU ERROR
 (5)  033000  104455                      TRAP    C$ERDF
 (6)  033002  000027                      .WORD   23
 (6)  033004  005212                      .WORD   EM23
 (6)  033006  007370                      .WORD   ERR23
8872  033010                        4$:   ESCAPE  SEG
 (3)  033010  104410                      TRAP    C$ESCAPE
 (3)  033012  000014                      .WORD   10000$-.
8873  033014  005202                      INC     R2              ;NEXT DATA
8874  033016  005205                      INC     R5              ;NEXT ADDRESS
8875  033020  022705  000010              CMP     #10,R5          ;DONE YET?
8876  033024  001324                      BNE     1$              ;BR IF NO
8877  033026                              ENDSEG
 (3)  033026                        10000$:
 (3)  033026  104405                      TRAP    C$ESEG
8878  033030                              EXIT    TST
 (3)  033030  104432                      TRAP    C$EXIT
 (3)  033032  000012                      .WORD   L10157-.
8879  033034     000     001     377  5$:   .BYTE   0,1,-1,0,0,253,125,0
      033037     000     000     253
      033042     125     000
8880
8881                                    .EVEN
8882  033044                            ENDTST
 (3)  033044                        L10157:
 (3)  033044  104401                      TRAP    C$ETST
8883
8884
8885  033046                            BADHEAD
 (2)                                    ;********** TEST 70 **********
8886                                    ;*ALU TEST
8887                                    ;*TEST OF ALU FUNCTION ADD W/C WITH C BIT SET
8888                                    ;*ALU FUNCTION (A PLUS B PLUS C)        CODE=01
8889                                    ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8890                                    ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
8891  033046                            BADHEAD
```

```
 (2)                                                ;******************** TEST 70 ********************
8892
8893  033046                                  BGNTST
 (3)  033046                                  T70::
8894  033046                                          MYINT
 (1)  033046  013701  002716                          MOV     KMCSR,R1                ;GET DEVICE ADDRESS.
8895  033052                                          MSTCLR                          ;MASTER CLEAR M8200,4,7
 (1)  033052  004537  003142                          JSR     R5,.MSTCLR                      ;CLEAR M8200,4,7
8896  033056  005005                                  CLR     R5                      ;MEM +SP ADDRESS
8897  033060  012702  033240                          MOV     #5$,R2                  ;POINTER TO CORRECT DATA
8898  033064  004737  003624                          JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
8899  033070  002654                                  MEMDAT                          ;POINTER TO DATA
8900  033072  004737  003776                          JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
8901  033076  002664                                  SPDAT                           ;POINTER TO DATA
8902  033100                                          BGNSEG
 (3)  033100  104404                                  TRAP    C$BSEG
8903  033102  004737  004062          1$:             JSR     PC,SETC                 ;SET C BIT.
8904  033106  042737  000017  033124                  BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
8905  033114  050537  033124                          BIS     R5,2$                   ;ADD ADDRESS TO INSTRUCTION
8906  033120                                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
 (1)  033120  004537  003230                          JSR     R5,.ROMCLK                      ;CLOCK INSTRUCTION
8907  033124  010000                  2$:             010000                          ;LOAD MAR
8908  033126  042737  000017  033144                  BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
8909  033134  050537  033144                          BIS     R5,3$                   ;ADD ADDRESS TO INSTRUCTION
8910  033140                                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  033140  004537  003230                          JSR     R5,.ROMCLK                      ;CLOCK INSTRUCTION
8911  033144  040420                  3$:             040400!<01*20>                  ;BR - ADD W/C
8912  033146                                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  033146  004537  003230                          JSR     R5,.ROMCLK                      ;CLOCK INSTRUCTION
8913  033152  061224                                  61224                           ;MOVE BR TO PORT4
8914  033154  111237  002636                          MOVB    (R2),$GDDAT                     ;PUT "EXPECTED" IN $GDDAT
8915  033160  116104  000004                          MOVB    4(R1),R4                ;PUT "FOUND" IN R4
8916  033164  123704  002636                          CMPB    $GDDAT,R4                       ;DATA CORRECT?
8917  033170  001411                                  BEQ     4$                      ;BR IF YES
8918  033172                                          ERROR   23,YES                  ;ALU ERROR
 (5)  033204  104455                                  TRAP    C$ERDF
 (6)  033206  000027                                  .WORD   23
 (6)  033210  005212                                  .WORD   EM23
 (6)  033212  007370                                  .WORD   ERR23
8919  033214                          4$:             ESCAPE  SEG
 (3)  033214  104410                                  TRAP    C$ESCAPE
 (3)  033216  000014                                  .WORD   10000$-.
8920  033220  005202                                  INC     R2                      ;NEXT DATA
8921  033222  005205                                  INC     R5                      ;NEXT ADDRESS
8922  033224  022705  000010                          CMP     #10,R5                  ;DONE YET?
8923  033230  001324                                  BNE     1$                      ;BR IF NO
8924  033232                                          ENDSEG
 (3)  033232                          10000$:
 (3)  033232  104405                                  TRAP    C$ESEG
8925  033234                                          EXIT    TST
 (3)  033234  104432                                  TRAP    C$EXIT
 (3)  033236  000012                                  .WORD   L10160-.
8926  033240     001     000     000  5$:             .BYTE   1,0,0,-1,253,0,0,125
      033243     377     253     000
      033246     000     125
8927
```

```
8928                                    .EVEN
8929  033250                            ENDTST
 (3)  033250                            L10160:
 (3)  033250  104401                            TRAP    C$ETST
8930
8931
8932  033252                                    BADHEAD
 (2)                                            ;********** TEST 71 **********
8933                                            ;*ALU TEST
8934                                            ;*TEST OF ALU FUNCTION SUB W/C WITH C BIT SET
8935                                            ;*ALU FUNCTION (A-B-C)  CODE=2
8936                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8937                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
8938  033252                                    BADHEAD
 (2)                                            ;********** TEST 71 **********
8939
8940
8941  033252                            BGNTST
 (3)  033252                            T71::
8942  033252                                    MYINT
 (1)  033252  013701  002716                      MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
8943  033256                                    MSTCLR                    ;MASTER CLEAR M8200,4,7
 (1)  033256  004537  003142                      JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
8944  033262  005005                            CLR     R5              ;MEM + SP ADDRESS
8945  033264  012702  033444                    MOV     #5$,R2          ;POINTER TO CORRECT DATA
8946  033270  004737  003624                    JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
8947  033274  002654                            MEMDAT                  ;POINTER TO DATA
8948  033276  004737  003776                    JSR     PC,SPLD.        ;LOAD 8 WORDS OF SP
8949  033302  002664                            SPDAT                   ;POINTER TO DATA
8950  033304                                    BGNSEG
 (3)  033304  104404                            TRAP    C$BSEG
8951  033306  004737  004062          1$:       JSR     PC,SETC         ;SET C BIT!
8952  033312  042737  000017  033330            BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
8953  033320  050537  033330                    BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
8954  033324                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  033324  004537  003230                      JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8955  033330  010000                  2$:       010000                  ;LOAD MAR
8956  033332  042737  000017  033350            BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
8957  033340  050537  033350                    BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
8958  033344                                    ROMCLK                  ;NEXT WORD IS INSTRUCITON, ROMCLK PC-55304
 (1)  033344  004537  003230                      JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8959  033350  040440                  3$:       040400!<2*20>           ;BR   SUB W/C
8960  033352                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  033352  004537  003230                      JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8961  033356  061224                            61224                   ;MOVE BR TO PORT4
8962  033360  111237  002636                    MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
8963  033364  116104  000004                    MOVB    4(R1),R4        ;PUT "FOUND" IN R.
8964  033370  123704  002636                    CMPB    $GDDAT,R4               ;DATA CORRECT?
8965  033374  001411                            BEQ     4$              ;BR IF YES
8966  033376                                    ERROR   23,YES          ;ALU ERROR
 (5)  033410  104455                            TRAP    C$ERDF
 (6)  033412  000027                            .WORD   23
 (6)  033414  005212                            .WORD   EM23
 (6)  033416  007370                            .WORD   ERR23
8967  033420                            4$:     ESCAPE  SEG
 (3)  033420  104410                            TRAP    C$ESCAPE
```

```
  (3)  033422  000014                          .WORD   10000$-.
8968   033424  005202                          INC     R2              ;NEXT DATA
8969   033426  005205                          INC     R5              ;NEXT ADDRESS
8970   033430  022705  000010                  CMP     #10,R5                  ;DONE YET?
8971   033434  001324                          BNE     1$              ;BR IF NO
8972   033436                                  ENDSEG
  (3)  033436                          10000$:
  (3)  033436  104405                          TRAP    C$ESEG
8973   033440                                  EXIT    TST
  (3)  033440  104432                          TRAP    C$EXIT
  (3)  033442  0000'2                          .WORD   L10161-.
8974   033444     000     001     377  5$:     .BYTE   0,1,-1,0,0,253,125,0
       033447     000     000     253
       033452     125     000
8975
8976                                    .EVEN
8977   033454                           ENDTST
  (3)  033454                           L10161:
  (3)  033454  104401                          TRAP    C$ETST
8978
8979
8980   033456                                  BADHEAD
  (2)                                           ;****************** TEST 72 ******************
8981                                            ;*ALU TEST
8982                                            ;*TEST OF ALU FUNCTION INC A WITH C  BIT SET
8983                                            ;*ALU FUNCTION (A PLUS 1)       CODE=3
8984                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8985                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
8986   033456                                  BADHEAD
  (2)                                           ;****************** TEST 72 ******************
8987
8988   033456                           BGNTST
  (3)  033456                           T72::
8989   033456                                  MYINT
  (1)  033456  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
8990   033462                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  033462  004537  003142                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
8991   033466  005005                          CLR     R5              ;MEM + SP ADDRESS
8992   033470  012702  033650                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
8993   033474  004737  003624                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
8994   033500  002654                          MEMDAT                  ;POINTER TO DATA
8995   033502  004737  003776                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
8996   033506  002664                          SPDAT                   ;POINTER TO DATA
8997   033510                                  BGNSEG
  (3)  033510  104404                          TRAP    C$BSEG
8998   033512  004737  004062          1$:     JSR     PC,SETC         ;SET C BIT!
8999   033516  042737  000017  033534          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
9000   033524  050537  033534                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
9001   033530                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  033530  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
9002   033534  010000                  2$:     010000                  ;LOAD MAR
9003   033536  042737  000017  033554          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
9004   033544  050537  033554                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
9005   033550                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  033550  004537  003230                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
9006   033554  040460                  3$:     040400.<3*20>           ;BR _ INC A
```

```
 9007  033556                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  033556  004537  003230                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 9008  033562  061224                            61224                   ;MOVE BR TO PORT4
 9009  033564  111237  002636                    MOVB    (R2),$GDDAT             ;PUT 'EXPECTED' IN $GDDAT
 9010  033570  116104  000004                    MOVB    4(R1),R4        ;PUT 'FOUND IN R4
 9011  033574  123704  002636                    CMPB    $GDDAT,R4               ;DATA CORRECT?
 9012  033600  001411                            BEQ     4$              ;BR IF YES
 9013  033602                                    ERROR   23,YES          ;ALU ERROR
  (5)  033614  104455                            TRAP    C$ERDF
  (6)  033616  000027                            .WORD   23
  (6)  033620  005212                            .WORD   EM23
  (6)  033622  007370                            .WORD   ERR23
 9014  033624                              4$:   ESCAPE  SEG
  (3)  033624  104410                            TRAP    C$ESCAPE
  (3)  033626  000014                            .WORD   10000$-.
 9015  033630  005202                            INC     R2              ;NEXT DATA
 9016  033632  005205                            INC     R5              ;NEXT ADDRESS
 9017  033634  022705  000010                    CMP     #10,R5          ;DONE YET?
 9018  033640  001324                            BNE     1$              ;BR IF NO
 9019  033642                                    ENDSEG
  (3)  033642                        10000$:
  (3)  033642  104405                            TRAP    C$ESEG
 9020  033644                                    EXIT    TST
  (3)  033644  104432                            TRAP    C$EXIT
  (3)  033646  000012                            .WORD   L10162-.
 9021  033650     001     001      000    5$:    .BYTE   1,1,0,0,126,126,253,253
       033653     000     126      126
       033656     253     253
 9022
 9023                                      .EVEN
 9024  033660                              ENDTST
  (3)  033660                              L10162:
  (3)  033660  104401                            TRAP    C$ETST
 9025
 9026
 9027  033662                                    BADHEAD
  (2)                                           ;*********************** TEST 73 ***********************
 9028                                           ;*ALU TEST
 9029                                           ;*TEST OF ALU FUNCTION 2A WITH C BIT SET
 9030                                           ;*ALU FUNCTION (A PLUS A)       CODE-5
 9031                                           ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 9032                                           ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 9033  033662                                    BADHEAD
  (2)                                           ;*********************** TEST 73 ***********************
 9034
 9035  033662                              BGNTST
  (3)  033662                              T73::
 9036  033662                                    MYINT
  (1)  033662  013701  002716                    MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 9037  033666                                    MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  033666  004537  003142                    JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 9038  033672  005005                            CLR     R5              ;MEM + SP ADDRESS
 9039  033674  012702  034054                    MOV     #5$,R2          ;POINTER TO CORRECT DATA
 9040  033700  004737  003624                    JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 9041  033704  002654                            MEMDAT                  ;POINTER TO DATA
 9042  033706  004737  003776                    JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
```

```
9043 033712 002664                          SPDAT                       ;POINTER TO DATA
9044 033714                                 BGNSEG
 (3) 033714 104404                          TRAP    C$BSEG
9045 033716 004737 004062          1$:      JSR     PC,SETC             ;SET C BIT!
9046 033722 042737 000017 033740            BIC     #17,2$              ;CLEAR ADDRESS FIELD OF INSTRUCTION
9047 033730 050537 033740                   BIS     R5,2$               ;ADD ADDRESS TO INSTRUCTION
9048 033734                                 ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1) 033734 004537 003230                   JSR     R5,.ROMCLK                  ;CLOCK INSTRUCTION
9049 033740 010000                 2$:      010000                      ;LOAD MAR
9050 033742 042737 000017 033760            BIC     #17,3$              ;CLEAR ADDRESS OF INSTRUCTION
9051 033750 050537 033760                   BIS     R5,3$               ;ADD ADDRESS TO INSTRUCTION
9052 033754                                 ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1) 033754 004537 003230                   JSR     R5,.ROMCLK                  ;CLOCK INSTRUCTION
9053 033760 040520                 3$:      040400!<5*20>               ;BR    2A
9054 033762                                 ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1) 033762 004537 003230                   JSR     R5,.ROMCLK                  ;CLOCK INSTRUCTION
9055 033766 061224                          61224                       ;MOVE BR TO PORT4
9056 033770 111237 002636                   MOVB    (R2),$GDDAT                 ;PUT 'EXPECTED'' IN $GDDAT
9057 033774 116104 000004                   MOVB    4(R1),R4            ;PUT 'FOUND IN R4
9058 034000 123704 002636                   CMPB    $GDDAT,R4                   ;DATA CORRECT?
9059 034004 001411                          BEQ     4$                  ;BR IF YES
9060 034006                                 ERROR   23,YES              ;ALU ERROR
 (5) 034020 104455                          TRAP    C$ERDF
 (6) 034022 000027                          .WORD   23
 (6) 034024 005212                          .WORD   EM23
 (6) 034026 007370                          .WORD   ERR23
9061 034030                        4$:      ESCAPE  SEG
 (3) 034030 104410                          TRAP    C$ESCAPE
 (3) 034032 000014                          .WORD   10000$-.
9062 034034 005202                          INC     R2                  ;NEXT DATA
9063 034036 005205                          INC     R5                  ;NEXT ADDRESS
9064 034040 022705 000010                   CMP     #10,R5              ;DONE YET?
9065 034044 001324                          BNE     1$                  ;BR IF NO
9066 034046                                 ENDSEG
 (3) 034046                        10000$:
 (3) 034046 104405                          TRAP    C$ESEG
9067 034050                                 EXIT    TST
 (3) 034050 104432                          TRAP    C$EXIT
 (3) 034052 000012                          .WORD   L10163-.
9068 034054    000    000    376   5$:      .BYTE   0,0,376,376,252,252,124,124
     034057    376    252    252
     034062    124    124
9069
9070                               .EVEN
9071 034064                        ENDTST
 (3) 034064                        L10163:
 (3) 034064 104401                          TRAP    C$ETST
9072
9073
9074 034066                                 BADHEAD
 (2)                                        ;******************* TEST 74 *******************
9075                                        ;*ALU TEST
9076                                        ;*TEST OF ALU FUNCTION A PLUS C WITH C BIT SET
9077                                        ;*ALU FUNCTION (A PLUS C)        CODE=4
9078                                        ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
9079                                        ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
```

```
 9080  034066                                          BADHEAD
  (2)                                                ;******************** TEST 74 ********************
 9081
 9082  034066                                    BGNTST
  (3)  034066                                    T74::
 9083  034066                                          MYINT
  (1)  034066  013701  002716                          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 9084  034072                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  034072  004537  003142                          JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 9085  034076  005005                                  CLR     R5              ;MEM + SP ADDRESS
 9086  034100  012702  034260                          MOV     #5$,R2          ;POINTER TO CORRECT DATA
 9087  034104  004737  003624                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 9088  034110  002654                                  MEMDAT                  ;POINTER TO DATA
 9089  034112  004737  003776                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 9090  034116  002664                                  SPDAT                   ;POINTER TO DATA
 9091  034120                                          BGNSEG
  (3)  034120  104404                                  TRAP    C$BSEG
 9092  034122  004737  004062                    1$:   JSR     PC,SETC         ;SET C BIT!
 9093  034126  042737  000017  034144                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 9094  034134  050537  034144                          BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 9095  034140                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  034140  004537  003230                          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 9096  034144  010000                              2$:  010000                  ;LOAD MAR
 9097  034146  042737  000017  034164                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 9098  034154  050537  034164                          BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 9099  034160                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  034160  004537  003230                          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 9100  034164  040500                              3$:  040400!<4*20>           ;BR   A PLUS C
 9101  034166                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  034166  004537  003230                          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 9102  034172  061224                                  61224                   ;MOVE BR TO PORT4
 9103  034174  111237  002636                          MOVB    (R2),$GDDAT             ;PUT 'EXPECTED'' IN $GDDAT
 9104  034200  116104  000004                          MOVB    4(R1),R4        ;PUT 'FOUND IN R4
 9105  034204  123704  002636                          CMPB    $GDDAT,R4               ;DATA CORRECT?
 9106  034210  001411                                  BEQ     4$              ;BR IF YES
 9107  034212                                          ERROR   23,YES          ;ALU ERROR
  (5)  034224  104455                                  TRAP    C$ERDF
  (6)  034226  000027                                  .WORD   23
  (6)  034230  005212                                  .WORD   EM23
  (6)  034232  007370                                  .WORD   ERR23
 9108  034234                                    4$:   ESCAPE  SEG
  (3)  034234  104410                                  TRAP    C$ESCAPE
  (3)  034236  000014                                  .WORD   10000$-.
 9109  034240  005202                                  INC     R2              ;NEXT DATA
 9110  034242  005205                                  INC     R5              ;NEXT ADDRESS
 9111  034244  022705  000010                          CMP     #10,R5          ;DONE YET?
 9112  034250  001324                                  BNE     1$              ;BR IF NO
 9113  034252                                          ENDSEG
  (3)  034252                                    10000$:
  (3)  034252  104405                                  TRAP     $ESEG
 9114  034254                                          EXIT    TST
  (3)  034254  104432                                  TRAP    C$EXIT
  (3)  034256  000012                                  .WORD   L10164-.
 9115  034260     001      001      000          5$:   .BYTE   1,1,0,0,126,126,253,253
       034263     000      126      126
       034266     253      253
```

```
 9116
 9117                                   .EVEN
 9118  034270                           ENDTST
  (3)  034270                           L10164:
  (3)  034270  104401                           TRAP    C$ETST
 9119
 9120  034272                                   BADHEAD
  (2)                                           ;********** TEST 75 **********
 9121                                           ;*ALU TEST
 9122                                           ;*TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT SET
 9123                                           ;*ALU FUNCTION (A-B-1)  CODE=17
 9124                                           ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 9125                                           ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 9126  034272                                   BADHEAD
  (2)                                           ;********** TEST 75 **********
 9127
 9128  034272                           BGNTST
  (3)  034272                           175::
```

```
 9130  034272
  (1)  034272  013701  002716          MYINT
 9131  034276                          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
  (1)  034276  004537  003142          MSTCLR                  ;MASTER CLEAR M8200,4,7
 9132  034302  005005                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 9133  034304  012702  034464          CLR     R5              ;MEM + SP ADDRESS
 9134  034310  004737  003624          MOV     #5$,R2          ;POINTER TO CORRECT DATA
 9135  034314  002654                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 9136  034316  004737  003776          MEMDAT                  ;POINTER TO DATA
 9137  034322  002664                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 9138  034324                          SPDAT                   ;POINTER TO DATA
  (3)  034324  104404                  BGNSEG
 9139  034326  004737  004062    1$:   TRAP    C$BSEG
 9140  034332  042737  000017  034350  JSR     PC,SETC         ;SET C BIT!
 9141  034340  050537  034350          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 9142  034344                          BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
  (1)  034344  004537  003230          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 9143  034350  010000          2$:     JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 9144  034352  042737  000017  034370  010000                  ;LOAD MAR
 9145  034360  050537  034370          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 9146  034364                          BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
  (1)  034364  004537  003230          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 9147  034370  040760          3$:     JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 9148  034372                          040400!<17*20>          ;BR   2'S COMP SUB
  (1)  034372  004537  003230          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 9149  034376  061224                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 9150  034400  111237  002636          61224                   ;MOVE BR TO PORT4
 9151  034404  116104  000004          MOVB    (R2),$GDDAT             ;PUT 'EXPECTED'' IN $GDDAT
 9152  034410  123704  002636          MOVB    4(R1),R4        ;PUT 'FOUND IN R4
 9153  034414  001411                  CMPB    $GDDAT,R4               ;DATA CORRECT?
 9154  034416                          BEQ     4$              ;BR IF YES
  (5)  034430  104455                  ERROR   23,YES          ;ALU ERROR
  (6)  034432  000027                  TRAP    C$ERDF
  (6)  034434  005212                  .WORD   23
  (6)  034436  007370                  .WORD   EM23
 9155  034440                          .WORD   ERR23
  (3)  034440  104410          4$:     ESCAPE  SEG
  (3)  034442  000014                  TRAP    C$ESCAPE
 9156  034444  005202                  .WORD   10000$-.
 9157  034446  005205                  INC     R2              ;NEXT DATA
 9158  034450  022705  000010          INC     R5              ;NEXT ADDRESS
 9159  034454  001324                  CMP     #10,R5          ;DONE YET?
 9160  034456                          BNE     1$              ;BR IF NO
  (3)  034456                          ENDSEG
  (3)  034456  104405          10000$:
 9161  034460                          TRAP    C$FSEG
  (3)  034460  104432                  EXIT    TST
  (3)  034462  000012                  TRAP    C$EXIT
 9162  034464     377     000     376  .WORD   L10165-.
       034467     377     377     252  5$:     .BYTE   -1,0,376,-1,-1,252,124,-1
       034472     124     377
 9163
 9164                                  .EVEN
 9165  034474                          ENDTST
  (3)  034474                          L10165:
  (3)  034474  104401                  TRAP    C$ETST
```

```
 9166
 9167
 9168  034476                                  BADHEAD
  (2)                                          ;********************* TEST 76 ********************
 9169                                          ;*ALU TEST
 9170                                          ;*TEST OF ALU FUNCTION DEC A WITH C BIT SE
 9171                                          ;*ALU FUNCTION (A-1)    CODE=7
 9172                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 9173                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 9174  034476                                  BADHEAD
  (2)                                          ;********************* TEST 76 ********************
 9175
 9176  034476                          BGNTST
  (3)  034476                          T76::
 9177  034476                                  MYINT
  (1)  034476  013701  002716                 MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 9178  034502                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  034502  004537  003142                 JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 9179  034506  005005                         CLR     R5              ;MEM + SP ADDRESS
 9180  034510  012702  034670                 MOV     #5$,R2          ;POINTER TO CORRECT DATA
 9181  034514  004737  003624                 JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 9182  034520  002654                         MEMDAT                  ;POINTER TO DATA
 9183  034522  004737  003776                 JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 9184  034526  002664                         SPDAT                   ;POINTER TO DATA
 9185  034530                                  BGNSEG
  (3)  034530  104404                         TRAP    C$BSEG
 9186  034532  004737  004062          1$:    JSR     PC,SETC         ;SET C BIT!
 9187  034536  042737  000017  034554         BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 9188  034544  050537  034554                 BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 9189  034550                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  034550  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 9190  034554  010000                  2$:    010000                  ;LOAD MAR
 9191  034556  042737  000017  034574         BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 9192  034564  050537  034574                 BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 9193  034570                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  034570  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 9194  034574  040560                  3$:    040400!<7*20>           ;BR   DEC A
 9195  034576                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  034576  004537  003230                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 9196  034602  061224                         61224                   ;MOVE BR TO PORT4
 9197  034604  111237  002636                 MOVB    (R2),$GDDAT             ;PUT 'EXPECTED' IN $GDDAT
 9198  034610  116104  000004                 MOVB    4(R1),R4        ;PUT 'FOUND IN R4
 9199  034614  123704  002636                 CMPB    $GDDAT,R4               ;DATA CORRECT?
 9200  034620  001411                         BEQ     4$              ;BR IF YES
 9201  034622                                  ERROR   23,YES          ;ALU ERROR
  (5)  034634  104455                         TRAP    C$ERDF
  (6)  034636  000027                         .WORD   23
  (6)  034640  005212                         .WORD   EM23
  (6)  034642  007370                         .WORD   ERR23
 9202  034644                          4$:    ESCAPE  SEG
  (3)  034644  104410                         TRAP    C$ESCAPE
  (3)  034646  000014                         .WORD   10000$-.
 9203  034650  005202                         INC     R2              ;NEXT DATA
 9204  034652  005205                         INC     R5              ;NEXT ADDRESS
 9205  034654  022705  000010                 CMP     #10,R5          ;DONE YET?
 9206  034660  001324                         BNE     1$              ;BR IF NO
```

```
 9207  034662                                        ENDSEG
  (3)  034662                                10000$:
  (3)  034662  104405                               TRAP    C$ESEG
 9208  034664                                        EXIT    TST
  (3)  034664  104432                               TRAP    C$EXIT
  (3)  034666  000012                               .WORD   L10166-.
 9209  034670     377     377     376    5$:       .BYTE   -1,-1,376,376,124,124,251,251
       034673     376     124     124
       034676     251     251
 9210
 9211                                       .EVEN
 9212  034700                               ENDTST
  (3)  034700                               L10166:
  (3)  034700  104401                               TRAP    C$ETST
 9213
 9214
```

```
9216
9217
9218
9219
9220
9221
```

```
                                .SBTTL  HARDWARE PARAMETER CODING SECTION



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

 9236  034702                           BGNHRD
  (3)  034702  000022                   .WORD L10167-L$HARD/2
  (3)  034704                   L$HARD::
 9237
 9238  034704                           GPRMD   WMP,0,0,7,0,7,YES
  (4)  034704  000032                   .WORD   T$CODE
  (4)  034706  034750                   .WORD   WMP
  (4)  034710  000007                   .WORD   7
  (4)  034712  000000                   .WORD   T$LOLIM
  (4)  034714  000007                   .WORD   T$HILIM
 9239  034716                           GPRMA   ADDRES,2,0,160000,177776,YES
  (4)  034716  001031                   .WORD   T$CODE
  (4)  034720  035027                   .WORD   ADDRES
  (4)  034722  160000                   .WORD   T$LOLIM
  (4)  034724  177776                   .WORD   T$HILIM
 9240  034726                           GPRMA   VECTOR,4,0,0,674,YES
  (4)  034726  002031                   .WORD   T$CODE
  (4)  034730  035066                   .WORD   VECTOR
  (4)  034732  000000                   .WORD   T$LOLIM
  (4)  034734  000674                   .WORD   T$HILIM
 9241  034736                           GPRMD   PRIRTY,6,0,7000,4,7,YES
  (4)  034736  003032                   .WORD   T$CODE
  (4)  034740  035130                   .WORD   PRIRTY
  (4)  034742  007000                   .WORD   7000
  (4)  034744  000004                   .WORD   T$LOLIM
  (4)  034746  000007                   .WORD   T$HILIM
 9242                           ;       GPRMD   LNUNIT,10,0,3,0,3,YES
 9243                           ;       GPRMD   SWPAC1,12,0,377,0,377,YES
 9244                           ;       GPRMD   SWPAC2,14,0,377,0,377,YES
 9245                           ;       GPRMD   LOOPBK,16,0,40000,0,1,YES
 9246
 9247  034750                           ENDHRD
  (2)                                   .EVEN
  (3)  034750                   L10167:
 9248
 9249  034750  044127  041511  020110  WMP:     .ASCIZ  'WHICH MICRO-PROCESSOR:''
       034756  044515  051103  026517
       034764  051120  041517  051505
       034772  047523  035122     000
 9250  034777     060  046475  031070           .ASCIZ  '0-M8200,4=M8204,7-M8207''
       035004  030060  032054  046475
       035012  031070  032060  033454
       035020  046475  031070  033460
```

```
                035026    000
 9251           035027    115  041511  047522  ADDRES: .ASCIZ  /MICRO-PROCESSOR CSR ADDRESS : /
                035034 050055  047522  042503
                035042 051523  051117  041440
                035050 051123  040440  042104
                035056 042522  051523  035040
                035064 000040
 9252           035066 044515  051103  026517  VECTOR: .ASCIZ  /MICRO-PROCESSOR VECTOR ADDRESS : /
                035074 051120  041517  051505
                035102 047523  020122  042526
                035110 052103  051117  040440
                035116 042104  042522  051523
                035124 035040  000040
 9253           035130 044515  051103  026517  PRIRTY: .ASCIZ  /MICRO-PROCESSOR PRIORITY LEVEL : /
                035136 051120  041517  051505
                035144 047523  020122  051120
                035152 047511  044522  054524
                035160 046040  053105  046105
                035166 035040  000040
 9254           035172 044127  041511  020110  LNUNIT: .ASCIZ  /WHICH LINE UNIT (0-3)? 0=NONE,1=M8201,2=M8202,3=M8203 : /
                035200 044514  042516  052440
                035206 044516  020124  030050
                035214 031455  037451  030040
                035222 047075  047117  026105
                035230 036461  034115  030062
                035236 026061  036462  034115
                035244 030062  026062  036463
                035252 034115  030062  020063
                035260 020072    000
 9255           035263    123  044527  041524  SWPAC1: .ASCIZ  /SWITCH PACK #1 (DDCMP LINE #) : /
                035270 020110  040520  045503
                035276 021440  020061  042050
                035304 041504  050115  046040
                035312 047111  020105  024443
                035320 035040  000040
 9256           035324 053523  052111  044103  SWPAC2: .ASCIZ  /SWITCH PACK #2 (BM873 BOOT ADR) : /
                035332 050040  041501  020113
                035340 031043  024040  046502
                035346 033470  020063  047502
                035354 052117  040440  051104
                035362 020051  020072    000
 9257           035367    127  046111  020114  LOOPBK: .ASCIZ  /WILL TEST CONNECTOR(S) BE USED ? 0=NO,1=YES : /
                035374 042524  052123  041440
                035402 047117  042516  052103
                035410 051117  051450  020051
                035416 042502  052440  042503
                035424 020104  020077  036460
                035432 047516  030454  054475
                035440 051505  035040  000040
 9258
 9259                                   .EVEN
 9260
 9261
 9262
 9263
 9264
```

```
9265
```

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

035446                                  BGNSFT
(3) 035446  000000                      .WORD L10170-L$SOFT/2
(3) 035450                      L$SOFT::

035450                                  ENDSFT
(2)                                     .EVEN
(3) 035450                      L10170:

                                .EVEN


035450                                  ENDMOD

035450                          CORMAX:

035450  000000                          .WORD   0               ;START OF NPR AREA (TEST 55)
        037776                  .-37776
037776  000000                  MEMEND: .WORD   0               ;END OF NPR AREA
040000                                  LASTAD
(2)                                     .EVEN
(4) 040000  000000                      .WORD   0
(4) 040002  000000                      .WORD   0
(3) 040004                      L$LAST::
        000114                          .LTN.ED=T$TESTNUM

                                        ; W A R N I N G < < < < <

                                        ;AREA BETWEEN CORMAX AND MEMEND USED BY TESTS IN DIAGNOSTIC.
                                        ; NO PATCHS OR DATA MY BE STORED IN THIS AREA.
                                        ;A SMALL PATCH AREA IS PROVIDED NEAR AREA 'DEBUG' FOR YOUR USE.
                                        ;ALSO THE AREA ABOVE ADDRESS 077776 MAY BE USED.
                                        ;
                                        ;ANYONE FOOLISH ENOUGH TO IGNOR THIS WARNING WILL BE DESTROYED.
                                        ;


        000001                  .END
```

```
ADDRES  035027            9239    9251#
ADR   = 000020 G          4953#
ASSEMB= 000010            4807
BIT0  = 000001 G          4953#   5224    5225    5934    7636
BIT00 = 000001 G          4953#
BIT01 = 000002 G          4953#
BIT02 = 000004 G          4953#
BIT03 = 000010 G          4953#
BIT04 = 000020 G          4953#
BIT05 = 000040 G          4953#
BIT06 = 000100 G          4953#
BIT07 = 000200 G          4953#
BIT08 = 000400 G          4953#
BIT09 = 001000 G          4953#
BIT1  = 000002 G          4953#   5222    5224    5225
BIT10 = 002000 G          4953#   5298    5306    6214    7470    7524
BIT11 = 004000 G          4953#   6001    6010
BIT12 = 010000 G          4953#   5710    6029    6038
BIT13 = 020000 G          4953#   5301    5309    5721
BIT14 = 040000 G          4953#
BIT15 = 100000 G          4953#   5725
BIT2  = 000004 G          4953#   5225
BIT3  = 000010 G          4953#
BIT4  = 000020 G          4953#
BIT5  = 000040 G          4953#
BIT6  = 000100 G          4953#   5207    5208
BIT7  = 000200 G          4953#   5208
BIT8  = 000400 G          4953#   5280    6251    7472    7526
BIT9  = 001000 G          4953#   5972    5981    6214    6249    6251    7470    7524
BOE   = 000400 G          4953#
CLKX    002570            5010#
CLRALL  003260            5228#
CLRC    004044            5411#   7730    7776    7822    7868    7914    7960    8006    8052    8098    8146    8194    8242
                          8289    8336    8383    8431
CORMAX  035450            7632    7664    9293#
CRAMA   003506            5307    5316#
CZDMP   002000 G          4813#
C$AU  = 000052            4807#   5820
C$AUTO= 000061            4807#   5774
C$BRK = 000022            4807#
C$BSEG- 000004            4807#   5896    5911    5935    5970    5980    5999    6009    6027    6037    6056    6094    6107
                          6134    6147    6174    6189    6213    6224    6275    6293    6324    6342    6373    6391    6422
                          6440    6471    6493    6528    6552    6589    6607    6638    6656    6687    6705    6736    6754
                          6785    6803    6834    6852    6883    6901    6932    6950    6981    7008    7038    7056    7107
                          7130    7173    7201    7690    7729    7775    7821    7867    7913    7959    8005    8051    8097
                          8145    8193    8241    8288    8335    8382    8430    8477    8526    8573    8620    8667    8714
                          8761    8808    8855    8902    8950    8997    9044    9091    9138    9185
C$BSUB= 000002            4807#   7106
C$CEFG= 000045            4807#
C$CLCK= 000062            4807#
C$CLEA= 000012            4807#   5787
C$CLOS= 000035            4807#
C$CLP1= 000006            4807#   7085
C$CVEC= 000036            4807#
C$DCLN= 000044            4807#
C$DODU= 000051            4807#   5770
```

```
C$DRPT= 000024   4807#
C$DU  = 000053   4807#  5804
C$EDIT= 000000   4807#  4848
C$ERDF= 000055   4807#  5383  5852  5869  5902  5916  5942  5948  5977  5985  6006  6014  6034
                 6042   6064  6070  6100  6114  6140  6155  6180  6194  6220  6229  6257  6285
                 6304   6334  6353  6383  6402  6432  6451  6484  6507  6542  6568  6599  6618
                 6648   6667  6697  6716  6746  6765  6795  6814  6844  6863  6893  6912  6942
                 6961   6999  7017  7048  7068  7084  7123  7148  7192  7213  7241  7243  7266
                 7268   7305  7338  7342  7376  7410  7443  7496  7537  7574  7584  7617  7653
                 7704   7745  7791  7837  7883  7929  7975  8021  8067  8113  8161  8209  8257
                 8304   8351  8398  8446  8493  8542  8589  8636  8683  8730  8777  8824  8871
                 8918   8966  9013  9060  9107  9154  9201
C$ERHR= 000056   4807#
C$ERRO= 000060   4807#
C$ERSF= 000054   4807#
C$ERSO= 000057   4807#
C$ESCA= 000010   4807#  5845  5856  5903  5917  5943  5949  5978  5986  6007  6035  6065  6071
                 6101   6115  6141  6156  6181  6195  6221  6230  6259  6286  6305  6335  6354
                 6384   6403  6433  6452  6485  6508  6544  6569  6600  6619  6649  6668  6698
                 6717   6747  6766  6796  6815  6845  6864  6894  6913  6943  6962  7000  7018
                 7049   7069  7124  7149  7193  7214  7377  7411  7444  7654  7705  7746  7792
                 7838   7884  7930  7976  8022  8068  8114  8162  8210  8258  8305  8352  8399
                 8447   8494  8543  8590  8637  8684  8731  8778  8825  8872  8919  8967  9014
                 9061   9108  9155  9202
C$ESEG= 000005   4807#  5908  5922  5952  5979  5987  6008  6015  6036  6043  6074  6105  6120
                 6145   6161  6188  6200  6222  6231  6290  6310  6339  6359  6388  6408  6437
                 6457   6490  6514  6549  6575  6604  6624  6653  6673  6702  6722  6751  6771
                 6800   6820  6849  6869  6898  6918  6947  6967  7005  7024  7053  7092  7128
                 7154   7198  7219  7706  7751  7797  7843  7889  7935  7981  8027  8073  8119
                 8167   8215  8263  8310  8357  8404  8452  8499  8548  8595  8642  8689  8736
                 8783   8830  8877  8924  8972  9019  9066  9113  9160  9207
C$ESUB= 000003   4807#  7159
C$ETST= 000001   4807#  5857  5871  5923  5953  5989  6017  6045  6075  6121  6162  6201  6232
                 6261   6311  6360  6409  6458  6515  6576  6625  6674  6723  6772  6821  6870
                 6919   6968  7025  7093  7160  7220  7246  7271  7308  7346  7381  7415  7448
                 7498   7542  7587  7619  7672  7707  7756  7802  7848  7894  7940  7986  8032
                 8078   8125  8172  8221  8268  8315  8362  8409  8457  8504  8553  8600  8647
                 8694   8741  8788  8835  8882  8929  8977  9024  9071  9118  9165  9212
C$EXIT- 000032   4807#  7378  7412  7445  7668  7752  7798  7844  7890  7936  7982  8028  8074
                 8120   8168  8216  8264  8311  8358  8405  8453  8500  8549  8596  8643  8690
                 8737   8784  8831  8878  8925  8973  9020  9067  9114  9161  9208
C$GETB= 000026   4807#
C$GETW= 000027   4807#
C$GMAN= 000043   4807#
C$GPHR= 000042   4807#  5675
C$GPLO= 000030   4807#
C$GPRI= 000040   4807#
C$INIT= 000011   4807#  5753
C$INLP= 000020   4807#
C$MANI= 000050   4807#
C$MEM = 000031   4807#
C$MSG - 000023   4807#  5556  5557  5558  5559  5560  5561  5562  5563  5564  5565  5566  5568
                 5570   5571  5572  5573  5574  5575  5576  5577  5578  5579  5580  5581  5582
                 5583   5584  5585  5586  5587  5588  5589  5594
C$OPEN= 000034   4807#
C$PNTB= 000014   4807#  5556  5557  5558  5559  5560  5561  5562  5563  5564  5565  5566  5568
```

```
                        5570    5571    5572    5573    5574    5575    5576    5577    5578    5579    5580    5581    5582
                        5583    5584    5585    5586    5587    5588    5589
C$PNTF= 000017          4807#   5592    5593
C$PNTS= 000016          4807#
C$PNTX= 000015          4807#
C$QIO = 000377          4807#
C$RDBU= 000007          4807#
C$REFG= 000047          4807#   5652    5655    5658    5662
C$RESE= 000033          4807#   5785    5803    7229    7304    7354
C$REVI= 000003          4807#   4848
C$RFLA= 000021          4807#
C$RPT = 000025          4807#   5624
C$SEFG= 000046          4807#
C$SPRI= 000041          4807#   7235    7239    7260    7264    7283    7298    7320    7336
C$SVEC= 000037          4807#
C$TPRI= 000013          4807#
DFPTBL  002364 G        4894#
DH0     005702          5499#   5566    5571    5572    5574    5575    5577    5584    5585
DH1     005703          5501#   5559    5560
DH2     005734          5502#   5556    5557    5558    5564    5565    5568    5573    5578    5579    5581    5583    5586
                        5587    5588    5589
DH27    006014          5505#   5580    5582
DH3     005750          5503#   5561    5562    5563
DH4     006000          5504#   5570    5576
DIAGMC= 000000          4807
EF.CON= 000036 G        4953#   5658
EF.NEW= 000035 G        4953#   5655
EF.PWR= 000034 G        4953#
EF.RES= 000037 G        4953#   5662
EF.STA= 000040 G        4953#   5652
EM1     004245          5455#   5592
EM10    004606          5467#
EM11    004640          5468#   7376    7410    7443    7496    7537    7653
EM12    004651          5469#
EM13    004673          5470#   7574    7584    7617
EM14    004715          5471#
EM15    004740          5472#   7745    7791    7837    7883    7929    7975    8021    8067    8113    8161    8209    8257
                        8304    8351    8398    8446    8589
EM16    005003          5474#
EM17    005050          5476#
EM2     004273          5456#   5916    5948    6180    6194
EM20    005077          5477#
EM21    005144          5478#
EM22    005173          5479#
EM23    005212          5480#   8493    8542    8636    8683    8730    8777    8824    8871    8918    8966    9013    9060
                        9107    9154    9201
EM24    005253          5481#
EM25    005437          5486#
EM26    005460          5487#   5869    5902    5977    5985    6006    6014    6034    6042    6220    6229
EM27    005511          5488#   5942    6064    6070    6100    6114    6140    6155    6285    6304    6334    6353    6383
                        6402    6432    6451    6484    6507    6542    6568
EM28    005540          5489#   6257
EM29    005571          5490#   6599    6618    6648    6667    6697    6716    6746    6765    6795    6814    6844    6863
                        6893    6912    6942    6961    6999
EM3     004402          5458#   7048    7068    7084
EM30    004337          5457#   7017
```

```
EM31    005276            5482#  7241  7266  7338
EM32    005325            5483#  7243  7268  7342
EM33    005364            5484#  7305
EM34    005421            5485#  7704
EM35    005050            5475#
EM36    005624            5383   5492#
EM37    005701            5493#  5852
EM4     004430            5460#  7123  7148  7192
EM5     004456            5462#  7213
EM6     004517            5464#
EM7     004545            5465#
ENDBUG  003230            5216#
ENDIT   011240            5659   5743  5747  5750  5752#
ERRFLG  002560            5006#
ERR1    006040 G          5556#
ERR10   006562 G          5563#
ERR11   006644 G          5564#  7376  7410  7443  7496  7537  7653
ERR12   006722 G          5565#
ERR13   007000 G          5566#  7574  7584  7617
ERR14   007026 G          5568#
ERR15   007104 G          5570#  7745  7791  7837  7883  7929  7975  8021  8067  8113  8161  8209  8257
                          8304   8351  8398  8446  8589
ERR16   007162 G          5571#
ERR17   007210 G          5572#
ERR2    006116 G          5557#  5916  5948  6180  6194
ERR20   007236 G          5573#
ERR21   007314 G          5574#
ERR22   007342 G          5575#
ERR23   007370 G          5576#  8493  8542  8636  8683  8730  8777  8824  8871  8918  8966  9013  9060
                          9107   9154  9201
ERR24   007446 G          5577#
ERR25   007474 G          5578#
ERR26   007552 G          5579#  5869  5902  5977  5985  6006  6014  6034  6042  6220  6229
ERR27   007630 G          5580#  5942  6064  6070  6100  6114  6140  6155  6285  6304  6334  6353  6383
                          6402   6432  6451  6484  6507  6542  6568
ERR28   007712 G          5581#  6257
ERR29   007770 G          5582#  6599  6618  6648  6667  6697  6716  6746  6765  6795  6814  6844  6863
                          6893   6912  6942  6961  6999
ERR3    006174 G          5558#  7048  7068  7084
ERR30   010052 G          5583#  7017
ERR31   010130 G          5584#  7241  7266  7338
ERR32   010156 G          5585#  7243  7268  7342
ERR33   010204 G          5586#  7305
ERR34   010262 G          5587#  7704
ERR35   010340 G          5588#
ERR36   010416 G          5383   5589#
ERR37   010474 G          5591#  5852
ERR4    006252 G          5559#  7123  7148  7192
ERR5    006334 G          5560#  7213
ERR6    006416 G          5561#
ERR7    006500 G          5562#
EVL     = 000004 G        4953#
E$END   = 002100          4807#
E$LOAD= 000035            4807#  4848
FLAG    002620            5023#
FM1     004100            5444#  5556  5557  5558  5559  5560  5561  5562  5563  5564  5565  5566  5568
```

```
                    5570  5571  5572  5573  5574  5575  5576  5577  5578  5579  5580  5581  5582
                    5583  5584  5585  5586  5587  5588  5589  5592
FTIME   002646      5034# 5644  5648*
F$AU  = 000015      4807# 5819  5820
F$AUTO= 000020      4807# 5756  5774
F$BGN = 000040      4807# 4813  4857  5556  5557  5558  5559  5560  5561  5562  5563  5564  5565
                    5566  5568  5570  5571  5572  5573  5574  5575  5576  5577  5578  5579  5580
                    5581  5582  5583  5584  5585  5586  5587  5588  5589  5591  5607  5615  5638
                    5756  5784  5801  5819  5838  5845  5856  5857  5863  5871  5891  5896  5903
                    5911  5917  5923  5931  5935  5943  5949  5953  5968  5970  5978  5980  5986
                    5989  5997  5999  6007  6009  6017  6025  6027  6035  6037  6045  6052  6056
                    6065  6071  6075  6089  6094  6101  6107  6115  6121  6129  6134  6141  6147
                    6156  6162  6170  6174  6181  6189  6195  6201  6209  6213  6221  6224  6230
                    6232  6245  6259  6261  6269  6275  6286  6293  6305  6311  6319  6324  6335
                    6342  6354  6360  6368  5373  6384  6391  6403  6409  6417  6422  6433  6440
                    6452  6458  6466  6471  6485  6493  6508  6515  6523  6528  6544  6552  6569
                    6576  6584  6589  6600  6607  6619  6625  6633  6638  6649  6656  6668  66 4
                    6682  6687  6698  6705  6717  6723  6731  6736  6747  6754  6766  6772  6780
                    6785  6796  6803  6815  6821  6829  6834  6845  6852  6864  6870  6878  6883
                    6894  6901  6913  6919  6927  6932  6943  6950  6962  6968  6976  6981  7000
                    7008  7018  7025  7033  7038  7049  7056  7069  7093  7101  7106  7107  7124
                    7130  7149  7159  7160  7168  7173  7193  7201  7214  7220  7227  7246  7253
                    7271  7279  7308  7316  7346  7353  7377  7378  7381  7388  7411  7412  7415
                    7422  7444  7445  7448  7456  7498  7510  7542  7551  7587  7595  7619  7627
                    7654  7668  7672  7683  7690  7705  7707  7720  7729  7746  7752  7756  7766
                    7775  7792  7798  7802  7812  7821  7838  7844  7848  7858  7867  7884  7890
                    7894  7904  7913  7930  7936  7940  7950  7959  7976  7982  7986  7996  8005
                    8022  8028  8032  8042  8051  8068  8074  8078  8088  8097  8114  8120  8125
                    8136  8145  8162  8168  8172  8183  8193  8210  8216  8221  8232  8241  8258
                    8264  8268  8279  8288  8305  8311  8315  8326  8335  8352  8358  8362  8373
                    8382  8399  8405  8409  8421  8430  8447  8453  8457  8468  8477  8494  8500
                    8504  8517  8526  8543  8549  8553  8564  8573  8590  8596  8600  8611  8620
                    8637  8643  8647  8658  8667  8684  8690  8694  8705  8714  8731  8737  8741
                    8752  8761  8778  8784  8788  8799  8808  8825  8831  8835  8846  8855  8872
                    8878  8882  8893  8902  8919  8925  8929  8941  8950  8967  8973  8977  8988
                    8997  9014  9020  9024  9035  9044  9061  9067  9071  9082  9091  9108  9114
                    9118  9128  9138  9155  9161  9165  9176  9185  9202  9208  9212  9236  9279
                    9290
F$CLEA= 000007      4807# 5784  5787
F$DU  = 000016      4807# 5801  5804
F$END = 000041      4807# 4813  5556  5557  5558  5559  5560  5561  5562  5563  5564  5565  5566
                    5568  5570  5571  5572  5573  5574  5575  5576  5577  5578  5579  5580  5581
                    5582  5583  5584  5585  5586  5587  5588  5589  5594  5615  5624  5753  5774
                    5787  5804  5820  5838  5845  5856  5857  5863  5871  5891  5903  5908  5917
                    5922  5923  5931  5943  5949  5952  5953  5968  5978  5979  5986  5987  5989
                    5997  6007  6008  6015  6017  6025  6035  6036  6043  6045  6052  6065  6071
                    6074  6075  6089  6101  6105  6115  6120  6121  6129  6141  6145  6156  6161
                    6162  6170  6181  6188  6195  6200  6201  6209  6221  6222  6230  6231  6232
                    6245  6259  6261  6269  6286  6290  6305  6310  6311  6319  6335  6339  6354
                    6359  6360  6368  6384  6388  6403  6408  6409  6417  6433  6437  6452  6457
                    6458  6466  6485  6490  6508  6514  6515  6523  6544  6549  6569  6575  6576
                    6584  6600  6604  6619  6624  6625  6633  6649  6653  6668  6673  6674  6682
                    6698  6702  6717  6722  6723  6731  6747  6751  6766  6771  6772  6780  6796
                    6800  6815  6820  6821  6829  6845  6849  6864  6869  6870  6878  6894  6898
                    6913  6918  6919  6927  6943  6947  6962  6967  6968  6976  7000  7005  7018
                    7024  7025  7033  7049  7053  7069  7092  7093  7101  7106  7124  7128  7149
```

```
                    7154  7159  7160  7168  7193  7198  7214  7219  7220  7227  7246  7253  7271
                    7279  7308  7316  7346  7353  7377  7378  7381  7388  7411  7412  7415  7422
                    7444  7445  7448  7456  7498  7510  7542  7551  7587  7595  7619  7627  7654
                    7668  7672  7683  7705  7706  7707  7720  7746  7751  7752  7756  7766  7792
                    7797  7798  7802  7812  7838  7843  7844  7848  7858  7884  7889  7890  7894
                    7904  7930  7935  7936  7940  7950  7976  7981  7982  7986  7996  8022  8027
                    8028  8032  8042  8068  8073  8074  8078  8088  8114  8119  8120  8125  8136
                    8162  8167  3168  8172  8183  8210  8215  8216  8221  8232  8258  8263  8264
                    8268  8279  8305  8310  8311  8315  8326  8352  8357  8358  8362  8373  8399
                    8404  8405  8409  8421  8447  8452  8453  8457  8468  8494  8499  8500  8504
                    8517  8543  8548  8549  8553  8564  8590  8595  8596  8600  8611  8637  8642
                    8643  8647  8658  8684  8689  8690  8694  8705  8731  8736  8737  8741  8752
                    8778  8783  8784  8788  8799  8825  8830  8831  8835  8846  8872  8877  8878
                    8882  8893  8919  8924  8925  8929  8941  8967  8972  8973  8977  8988  9014
                    9019  9020  9024  9035  9061  9066  9067  9071  9082  9108  9113  9114  9118
                    9128  9155  9160  9161  9165  9176  9202  9207  9208  9212  9247  9282  9290
F$HARD= 000004       4807#  9236  9247
F$HW  = 000013       4807#  4894  4909
F$INIT= 000006       4807#  5638  5753
F$JMP = 000050       4807#  5615  7378  7412  7445  7668  7752  7798  7844  7890  7936  7982  8028
                     8074   8120  8168  8216  8264  8311  8358  8405  8453  8500  8549  8596  8643
                     8690   8737  8784  8831  8878  8925  8973  9020  9067  9114  9161  9208
F$MOD - 000000       4807#  4813  9290
F$MSG = 000011       4807#  5556  5557  5558  5559  5560  5561  5562  5563  5564  5565  5566  5568
                     5570   5571  5572  5573  5574  5575  5576  5577  5578  5579  5580  5581  5582
                     5583   5584  5585  5586  5587  5588  5589  5591  5594
F$PROT  000021       4807#  4857  4861
F$PWR = 000017       4807#
F$RPT = 000012       4807#  5607  5624
F$SEG = 000003       4807#  5896  5908  5911  5922  5935  5952  5970  5979  5980  5987  5999  6008
                     6009   6015  6027  6036  6037  6043  6056  6074  6094  6105  6107  6120  6134
                     6145   6147  6161  6174  6188  6189  6200  6213  6222  6224  6231  6275  6290
                     6293   6310  6324  6339  6342  6359  6373  6388  6391  6408  6422  6437  6440
                     6457   6471  6490  6493  6514  6528  6549  6552  6575  6589  6604  6607  6624
                     6638   6653  6656  6673  6687  6702  6705  6722  6736  6751  6754  6771  6785
                     6800   6803  6820  6834  6849  6852  6869  6883  6898  6901  6918  6932  6947
                     6950   6967  6981  7005  7008  7024  7038  7053  7056  7092  7107  7128  7130
                     7154   7173  7198  7201  7219  7690  7706  7729  7751  7775  7797  7821  7843
                     7867   7889  7913  7935  7959  7981  8005  8027  8051  8073  8097  8119  8145
                     8167   8193  8215  8241  8263  8288  8310  8335  8357  8382  8404  8430  8452
                     8477   8499  8526  8548  8573  8595  8620  8642  8667  8689  8714  8736  8761
                     8783   8808  8830  8855  8877  8902  8924  8950  8972  8997  9019  9044  9066
                     9091   9113  9138  9160  9185  9207
F$SOFT- 000005       4807#  9279  9282
F$SRV = 000010       4807#
F$SUB = 000002       4807#  7106  7159
F$SW  - 000014       4807#  4923  4926
F$TEST= 000001       4807#  5838  5857  5863  5871  5891  5923  5931  5953  5968  5989  5997  6017
                     6025   6045  6052  6075  6089  6121  6129  6162  6170  6201  6209  6232  6245
                     6261   6269  6311  6319  6360  6368  6409  6417  6458  6466  6515  6523  6576
                     6584   6625  6633  6674  6682  6723  6731  6772  6780  6821  6829  6870  6878
                     6919   6927  6968  6976  7025  7033  7093  7101  7160  7168  7220  7227  7246
                     7253   7271  7279  7308  7316  7346  7353  7381  7388  7415  7422  7448  7456
                     7498   7510  7542  7551  7587  7595  7619  7627  7672  7683  7707  7720  7756
                     7766   7802  7812  7848  7858  7894  7904  7940  7950  7986  7996  8032  8042
                     8078   8088  8125  8136  8172  8183  8221  8232  8268  8279  8315  8326  8362
```

```
                        8373    8409    8421    8457    8468    8504    8517    8553    8564    8600    8611    8647    8658
                        8694    8705    8741    8752    8788    8799    8835    8846    8882    8893    8929    8941    8977
                        8988    9024    9035    9071    9082    9118    9128    9165    9176    9212
GETPRM  010676           5663    5671#   5676
G$CNTO= 000200          4807#
G$DISP= 000003          4807#
G$EXCP= 000400          4807#
G$HILI= 000002          4807#
G$LOLI= 000001          4807#
G$NO  = 000000          4807#
G$OFFS= 000400          4807#   9238    9239    9240    9241
G$OFSI= 000376          4807#   9238    9239    9240    9241
G$PRMA= 000001          4807#   9239    9240
G$PRMD= 000002          4807#   9238    9241
G$PRML= 000000          4807#
G$RADA= 000140          4807#
G$RADB= 000000          4807#
G$RADD= 000040          4807#
G$RADL= 000120          4807#
G$RADO= 000020          4807#   9238    9239    9240    9241
G$XFER= 000004          4807#
G$YES = 000010          4807#   9238    9239    9240    9241
HELP  = 000000          4794#   4840    4851    4874    5123    5609    5617
HOE   = 100000 G        4953#
IBE   = 010000 G        4953#
IDU   = 000040 G        4953#
IER   = 020000 G        4953#
INIFLG  002674          5048#
INSTU   003522          5308    5318#
ISR   = 000100 G        4953#
IXE   = 004000 G        4953#
I$AU  = 000041          4807#   5819#   5820#
I$AUTO= 000041          4807#   5756#   5774#
I$CLN = 000041          4807#   5784#   5787#
I$DU  = 000041          4807#   5801#   5804#
I$HRD = 000041          9236#   9247#
I$INIT= 000041          4807#   5638#   5753#
I$MOD = 000041          4807#   4813#   9290#
I$MSG = 000041          4807#   5556#   5557#   5558#   5559#   5560#   5561#   5562#   5563#   5564#   5565#   5566#   5568#
                        5570#   5571#   5572#   5573#   5574#   5575#   5576#   5577#   5578#   5579#   5580#   5581#   5582#
                        5583#   5584#   5585#   5586#   5587#   5588#   5589#   5591#   5594#
I$PROT= 000040          4807#   4857#
I$PTAB= 000041          4807#
I$PWR = 000041          4807#
I$RPT = 000041          4807#   5607#   5624#
I$SEG = 000041          4807#   5838    5863    5891    5896#   5903    5908#   5911#   5917    5922#   5931    5935#   5943
                        5949    5952#   5968    5970#   5978    5979#   5980#   5986    5987#   5997    5999#   6007    6008#
                        6009#   6015#   6025    6027#   6035    6036#   6037#   6043#   6052    6056#   6065    6071    6074#
                        6089    6094#   6101    6105#   6107#   6115    6120#   6129    6134#   6141    6145#   6147#   6156
                        6161#   6170    6174#   6181    6188#   6189#   6195    6200#   6209    6213#   6221    6222#   6224#
                        6230    6231#   6245    6269    6275#   6286    6290#   6293#   6305    6310#   6319    6324#   6335
                        6339#   6342#   6354    6359#   6368    6373#   6384    6388#   6391#   6403    6408#   6417    6422#
                        6433    6437#   6440#   6452    6457#   6466    6471#   6485    6490#   6493#   6508    6514#   6523
                        6528#   6544    6549#   6552#   6569    6575#   6584    6589#   6600    6604#   6607#   6619    6624#
                        6633    6638#   6649    6653#   6656#   6668    6673#   6682    6687#   6698    6702#   6705#   6717
                        6722#   6731    6736#   6747    6751#   6754#   6766    6771#   6780    6785#   6796    6800#   6803#
```

```
                    6815    6820#   6829    6834#   6845    6849#   6852#   6864    6869#   6878    6883#   6894    6898#
                    6901#   6913    6918#   6927    6932#   6943    6947#   6950#   6962    6967#   6976    6981#   7000
                    7005#   7008#   7018    7024#   7033    7038#   7049    7053#   7056#   7069    7092#   7101    7106
                    7107#   7124    7128#   7130#   7154#   7168    7173#   7193    7198#   7201#   7214    7219#   7227
                    7253    7279    7316    7353    7388    7422    7456    7510    7551    7595    7627    7683    7690#
                    7705    7706#   7720    7729#   7746    7751#   7766    7775#   7792    7797#   7812    7821#   7838
                    7843#   7858    7867#   7884    7889#   7904    7913#   7930    7935#   7950    7959#   7976    7981#
                    7996    8005#   8022    8027#   8042    8051#   8068    8073#   8088    8097#   8114    8119#   8136
                    8145#   8162    8167#   8183    8193#   8210    8215#   8232    8241#   8258    8263#   8279    8288#
                    8305    8310#   8326    8335#   8352    8357#   8373    8382#   8399    8404#   8421    8430#   8447
                    8452#   8468    8477#   8494    8499#   8517    8526#   8543    8548#   8564    8573#   8590    8595#
                    8611    8620#   8637    8642#   8658    8667#   8684    8689#   8705    8714#   8731    8736#   8752
                    8761#   8778    8783#   8799    8808#   8825    8830#   8846    8855#   8872    8877#   8893    8902#
                    8919    8924#   8941    8950#   8967    8972#   8988    8997#   9014    9019#   9035    9044#   9061
                    9066#   9082    9091#   9108    9113#   9128    9138#   9155    9160#   9176    9185#   9202    9207#
I$SETU= 000041       4807#
I$SFT  = 000041     9279#   9282#
I$SRV  = 000041      4807#
I$SUB  = 000041     4807#   5838    5863    5891    5931    5968    5997    6025    6052    6089    6129    6170    6209
                    6245    6269    6319    6368    6417    6466    6523    6584    6633    6682    6731    6780    6829
                    6878    6927    6976    7033    7101    7106#   7159#   7168    7227    7253    7279    7316    7353
                    7388    7422    7456    7510    7551    7595    7627    7683    7720    7766    7812    7858    7904
                    7950    7996    8042    8088    8136    8183    8232    8279    8326    8373    8421    8468    8517
                    8564    8611    8658    8705    8752    8799    8846    8893    8941    8988    9035    9082    9128
                    9176
I$TST  = 000041     4807#   5838#   5845    5856    5857#   5863#   5871#   5891#   5923#   5931#   5953#   5968#   5989#
                    5997#   6017#   6025#   6045#   6052#   6075#   6089#   6121#   6129#   6162#   6170#   6201#   6209#
                    6232#   6245#   6259    6261#   6269#   6311#   6319#   6360#   6368#   6409#   6417#   6458#   6466#
                    6515#   6523#   6576#   6584#   6625#   6633#   6674#   6682#   6723#   6731#   6772#   6820#   6821#
                    6829#   6870#   6878#   6919#   6927#   6968#   6976#   7025#   7033#   7093#   7101#   7106    7149
                    7160#   7168#   7220#   7227#   7246#   7253#   7271#   7279#   7308#   7316#   7346#   7353#   7377
                    7378    7381#   7388#   7411    7412    7415#   7422#   7444    7445    7448#   7456#   7498#   7510#
                    7542#   7551#   7587#   7595#   7619#   7627#   7654    7668    7672#   7683#   7707#   7720#   7752
                    7756#   7766#   7798    7802#   7812#   7844    7848#   7858#   7890    7894#   7904#   7936    7940#
                    7950#   7982    7986#   7996#   8028    8032#   8042#   8074    8078#   8088#   8120    8125#   8136#
                    8168    8172#   8183#   8216    8221#   8232#   8264    8268#   8279#   8311    8315#   8326#   8358
                    8362#   8373#   8405    8409#   8421#   8453    8457#   8468#   8500    8504#   8517#   8549    8553#
                    8564#   8596    8600#   8611#   8643    8647#   8658#   8690    8694#   8705#   8737    8741#   8752#
                    8784    8788#   8799#   8831    8835#   8846#   8878    8882#   8893#   8925    8929#   8941#   8973
                    8977#   8988#   9020    9024#   9035#   9067    9071#   9082#   9114    9118#   9128#   9161    9165#
                    9176#   9208    9212#
J$JMP  = 000167     4807#   5615
KMACTV  002610      5019#
KMCSR   002716      5084#   5679*   5758    5839    5864    5892    5909    5936    5971    6000    6028    6171    6186
                    6212    6246    6274    6323    6372    6421    6470    6527    6588    6637    6686    6735    6784
                    6833    6882    6931    6980    7037    7102    7172    7228    7254    7280    7317    7356    7389
                    7423    7458    7511    7552    7596    7628    7684    7721    7767    7813    7859    7905    7951
                    7997    8043    8089    8137    8184    8233    8280    8327    8374    8422    8469    8518    8565
                    8612    8659    8706    8753    8800    8847    8894    8942    8989    9036    9083    9130    9177
KMCSRH  002720      5085#   5207*   5208*   5222*   5224*   5225*   5681*   5682*
KMCTL   002722      5086#   5684*   5685*   6058
KMNUM   002612      5020#
KMPO4   002724      5087#   5366*   5380    5400*   5687*   5688*   6092
KMPO6   002726      5088#   5223*   5690*   5691*   6132    7459    7460
KMRLVL  002710      5081#   5330*   5695*   5696*
KMRVEC  002706      5080#   5328*   5693*
```

```
KMTLVL  002714            5083#   5331*   5701*   5702*
KMTVEC  002712            5082#   5329*   5698*   5699*
LNUNIT  035172            9254#
LOCK    002442            4989#
LOE   = 040000 G          4953#
LOGDEV  002552            5003#   5666*   5672*   5673    5675    5770
LOKFLG  002676            5050#
LOOPBK  035367            9257#
LOT   = 000010 G          4953#
LTN.ED= 000114            9299#
L$ACP   002110 G          4848#
L$APT   002036 G          4848#
L$AU    001350 G          4848    5819#
L$AUT   002070 G          4848#
L$AUTO  011242 G          4848    5756#
L$CCP   002106 G          4848#
L$CLEA  011340 G          4848    5784#
L$CO    002032 G          4848#
L$DEPO  002011 G          4848#
L$DESC  002414 G          4849    4983#
L$DESP  002076 G          48 8#
L$DEVP  002060 G          4848#
L$DISP  002132 G          4848    4871#
L$DLY   002116 G          4848#
L$DTP   002040 G          4848#
L$DTYP  002034 G          4848#
L$DU    011344 G          4848    5801#
L$DUT   002072 G          4848#
L$DVTY  003130 G          4848    5116#
L$EF    002052 G          4848#
L$ETP   002102 G          4848#
L$EXP1  002042 G          4848#
L$EXP2  002044 G          4848#
L$EXP3  002046 G          4848#
L$EXP4  002064 G          4848#
L$EXP5  002066 G          4848#
L$HARD  034704 G          4848    9236#
L$HIME  002120 G          4848#
L$HPCP  002016 G          4848#
L$HPTP  002022 G          4848#
L$HW    002364 G          4848    4894#
L$ICP   002104 G          4848#
L$INIT  010554 G          4848    5638#
L$LADP  002026 G          4848#
L$LAST  040004 G          4848    9298#
L$LOAD  002100 G          4848#
L$LUN   002074 G          4848#
L$MREV  002050 G          4848#
L$NAME  002000 G          4848#
L$PROT  002122 G          4848    4857#
L$PRT   002112 G          4848#
L$REPP  002062 G          4848#
L$REV   002010 G          4848#
L$RPT   010546 G          5607#
L$SOFT  035450 G          9279#
L$SPC   002056 G          4848#
```

```
L$SPCP  002020 G          4848#
L$SPTP  002024 G          4848#
L$STA   002030 G          4848#
L$SW    002414 G          4923#
L$TEST  002114 G          4848#
L$TIML  002014 G          4848#
L$UNIT  002012 G          4848#   5673
L10001  002412            4894    4909#
L10002  002414            4923    4926#
L10003  006114            5556#
L10004  006172            5557#
L10005  006250            5558#
L10006  006332            5559#
L10007  006414            5560#
L10010  006476            5561#
L10011  006560            5562#
L10012  006642            5563#
L10013  006720            5564#
L10014  006776            5565#
L10015  007024            5566#
L10016  007102            5568#
L10017  007160            5570#
L10020  007206            5571#
L10021  007234            5572#
L10022  007312            5573#
L10023  007340            5574#
L10024  007366            5575#
L10025  007444            5576#
L10026  007472            5577#
L10027  007550            5578#
L10030  007626            5579#
L10031  007710            5580#
L10032  007766            5581#
L10033  010050            5582#
L10034  010126            5583#
L10035  010154            5584#
L10036  010202            5585#
L10037  010260            5586#
L10040  010336            5587#
L10041  010414            5588#
L10042  010472            5589#
L10043  010544            5594#
L10044  010552            5615    5624#
L10045  011240            5753#
L10046  011336            5774#
L10047  011342            5787#
L10050  011346            5804#
L10051  011350            5820#
L10052  011476            5845    5856    5857#
L10053  011542            5871#
L10054  011730            5923#
L10055  012074            5953#
L10056  012230            5989#
L10057  012360            6017#
L10060  012510            6045#
L10061  012652            6075#
```

L10062	013036	6121#		
L10063	013222	6162#		
L10064	013410	6201#		
L10065	013560	6232#		
L10066	013666	6259	6261#	
L10067	014116	6311#		
L10070	014346	6360#		
L10071	014576	6409#		
L10072	015026	6458#		
L10073	015312	6515#		
L10074	015606	6576#		
L10075	016036	6625#		
L10076	016266	6674#		
L10077	016516	6723#		
L10100	016746	6772#		
L10101	017176	6821#		
L10102	017426	6870#		
L10103	017656	6919#		
L10104	020106	6968#		
L10105	020404	7025#		
L10106	020724	7093#		
L10107	021270	7149	7160#	
L10110	021266	7159#		
L10111	021602	7220#		
L10112	021744	7246#		
L10113	022106	7271#		
L10114	022262	7308#		
L10115	022466	7346#		
L10116	022632	7377	7378	7381#
L10117	023002	7411	7412	7415#
L10120	023146	7444	7445	7448#
L10121	023334	7498#		
L10122	023550	7542#		
L10123	023766	7587#		
L10124	024120	7619#		
L10125	024336	7654	7668	7672#
L10126	024466	7707#		
L10127	024702	7752	7756#	
L10130	025106	7798	7802#	
L10131	025312	7844	7848#	
L10132	025516	7890	7894#	
L10133	025722	7936	7940#	
L10134	026126	7982	7986#	
L10135	026332	8028	8032#	
L10136	026536	8074	8078#	
L10137	026744	8120	8125#	
L10140	027150	8168	8172#	
L10141	027354	8216	8221#	
L10142	027560	8264	8268#	
L10143	027764	8311	8315#	
L10144	030170	8358	8362#	
L10145	030374	8405	8409#	
L10146	030600	8453	8457#	
L10147	031004	8500	8504#	
L10150	031210	8549	8553#	
L10151	031414	8596	8600#	

```
L10152  031620          8643    8647#
L10153  032024          8690    8694#
L10154  032230          8737    8741#
L10155  032434          8784    8788#
L10156  032640          8831    8835#
L10157  033044          8878    8882#
L10160  033250          8925    8929#
L10161  033454          8973    8977#
L10162  033660          9020    9024#
L10163  034064          9067    9071#
L10164  034270          9114    9118#
L10165  034474          9161    9165#
L10166  034700          9208    9212#
L10167  034750          9236    9247#
L10170  035450          9279    9282#
MASKX   002572          5011#
MDATA   002506          4996#
MEMDAT  002654          5041#   7726    7772    7818    7864    7910    7956    8002    8048    8094    8142    8190    8238
                        8285    8332    8379    8427    8474    8523    8570    8617    8664    8711    8758    8805    8852
                        8899    8947    8994    9041    9088    9135    9182
MEMEND  037776          5016    9297#
MEMLD   003624          5353#   7686    7725    7771    7817    7863    7909    7955    8001    8047    8093    8141    8189
                        8237    8284    8331    8378    8426    8473    8522    8569    8616    8663    8710    8757    8804
                        8851    8898    8946    8993    9040    9087    9134    9181
MEMLD2  003634          5360#
MEMLIM  002604          5016#   7662
MEMSET  003410          5289#
MEMSZ   002606          5017#   5740*   5748*
MRO     002624          5025#   5383    5580    5582    5852    5869    5902    5916    5933*   5942    5948    5977    5985
                        6006    6014    6034    6042    6054*   6064    6070    6090*   6100    6114    6130*   6140    6155
                        6180    6194    6220    6229    6257    6271*   6321*   6370*   6419*   6468*   6525*   6586*   6635*
                        6684*   6733*   6782*   6831*   6880*   6929*   7241    7243    7266    7268    7305    7338    7342
                        7376    7410    7443    7496    7537    7574    7584    7617    7653    7704    7745    7791    7837
                        7883    7929    7975    8021    8067    8113    8161    8209    8257    8304    8351    8398    8446
                        8493    8542    8589    8636    8683    8730    8777    8824    8871    8918    8966    9013    9060
                        9107    9154    9201
NEWST   010670          5653    5656    5664#   5674
NEXT    002440          4988#
NPRSET  003560          5335#   7359    7392    7426    7461    7518    7554    7598    7639
ONE     002602          5015#
O$APTS= 000000          4807#   4848
O$AU  = 000001          4807#   4838#   4848
O$BGNR= 000000          4807#   4848
O$BGNS= 000000          4807#   4848
O$DU  = 000001          4807#   4838#   4848
O$ERRT= 000000          4807#   4848
O$GNSW= 000000          4807#   4848
O$POIN= 000001          4807#   4838#   4848
O$SETU= 000000          4807#   4848    9298
PNT     001000  G       4953#
POP2SP= 022626          4962#
PRI   = 002000  G       4953#
PRIRTY  035130          9241    9253#
PRI00 - 000000  G       4953#   7239    7264
PRI01 = 000040  G       4953#
PRI02 - 000100  G       4953#
```

```
PRIO3 = 000140 G          4953#
PRIO4 = 000200 G          4953#
PRIO5 = 000240 G          4953#
PRIO6 = 000300 G          4953#
PRIO7 = 000340 G          4953#  7235    7260
PSTACK  002554            5004#  5643*
QV.FLG  002677            5051#
RAMDAT  003354            5273#
REGADR  002730            5092#
RETADR  002562            5007#
RUN     002622            5024#
SAVACT  002614            5021#
SAVE4   002650            5035#  5646*   5649    5772    5854    7541
SAVE6   002652            5036#  5647*   5650    5773    5855    7540
SAVNUM  002616            5022#
SAVPC   002576            5013#
SAVSP   002574            5012#
SETBR0  003304            5238#
SETBR1  003314            5244#
SETBR4  003324            5251#
SETBR7  003334            5258#
SETC    004062            5421#  8478    8527    8574    8621    8668    8715    8762    8809    8856    8903    8951    8998
                          9045   9092    9139    9186
SETVEC  003536            5325#  7231    7256    7291    7329
SETZ    003344            5266#
SFPTBL  002414 G          4923#
SPDAT   002664            5042#  7728    7774    7820    7866    7912    7958    8004    8050    8096    8144    8192    8240
                          8287   8334    8381    8429    8476    8525    8572    8619    8666    8713    8760    8807    8854
                          8901   8949    8996    9043    9090    9137    9184
SPLD    003776            5393#  7688    7727    7773    7819    7865    7911    7957    8003    8049    8095    8143    8191
                          8239   8286    8333    8380    8428    8475    8524    8571    8618    8665    8712    8759    8806
                          6853   8900    8948    8995    9042    9089    9136    9183
SSTACK  003130            5096#  5641
STAT    002566            5009#
STAT1   002700            5073#  5704*   5706*   5710*   5721*   5725*   5728*   7284    7321
STAT2   002702            5074#  5731*   5733*
STAT3   002704            5075#
STOP    023346            7513#
STRTSW  002564            5008#
SUBRPC  002556            5005#
SVCGBL= 000000            4807#  4813    4821#   4848    4857    4871    4894    4923    4983    5116    5556    5557    5558
                          5559   5560    5561    5562    5563    5564    5565    5566    5568    5570    5571    5572    5573
                          5574   5575    5576    5577    5578    5579    5580    5581    5582    5583    5584    5585    5586
                          5587   5588    5589    5591    5607    5638    5756    5784    5801    5819    9236    9279    9298#
SVCINS- 000000            4807#  4818#   4848    4871    4894    4923    4983    5116    5383    5556    5557    5558    5559
                          5560   5561    5562    5563    5564    5565    5566    5568    5570    5571    5572    5573    5574
                          5575   5576    5577    5578    5579    5580    5581    5582    5583    5584    5585    5586    5587
                          5588   5589    5592    5593    5594    5615    5624    5652    5653    5655    5656    5658    5659
                          5662   5663    5675    5676    5753    5770    5774    5785    5787    5803    5804    5820    5845
                          5852   5856    5857    5869    5871    5896    5902    5903    5908    5911    5916    5917    5922
                          5923   5935    5942    5943    5948    5949    5952    5953    5970    5977    5978    5979    5980
                          5985   5986    5987    5989    5999    6006    6007    6008    6009    6014    6015    6017    6027
                          6034   6035    6036    6037    6042    6043    6045    6056    6064    6065    6070    6071    6074
                          6075   6094    6100    6101    6105    6107    6114    6115    6120    6121    6134    6140    6141
                          6145   6147    6155    6156    6161    6162    6174    6180    6181    6188    6189    6194    6195
                          6200   6201    6213    6220    6221    6222    6224    6229    6230    6231    6232    6257    6259
```

```
                6261  6275  6285  6286  6290  6293  6304  6305  6310  6311  6324  6334  6335
6339  6342  6353  6354  6359  6360  6373  6383  6384  6388  6391  6402  6403
6408  6409  6422  6432  6433  6437  6440  6451  6452  6457  6458  6471  6484
6485  6490  6493  6507  6508  6514  6515  6528  6542  6544  6549  6552  6568
6569  6575  6576  6589  6599  6600  6604  6607  6618  6619  6624  6625  6638
6648  6649  6653  6656  6667  6668  6673  6674  6687  6697  6698  6702  6705
6716  6717  6722  6723  6736  6746  6747  6751  6754  6765  6766  6771  6772
6785  6795  6796  6800  6803  6814  6815  6820  6821  6834  6844  6845  6849
6852  6863  6864  6869  6870  6883  6893  6894  6898  6901  6912  6913  6918
6919  6932  6942  6943  6947  6950  6961  6962  6967  6968  6981  6999  7000
7005  7008  7017  7018  7024  7025  7038  7048  7049  7053  7056  7068  7069
7084  7085  7092  7093  7106  7107  7123  7124  7128  7130  7148  7149  7154
7159  7160  7173  7192  7193  7198  7201  7213  7214  7219  7220  7229  7235
7239  7241  7243  7246  7260  7264  7266  7268  7271  7283  7298  7304  7305
7308  7320  7336  7338  7342  7346  7354  7376  7377  7378  7381  7410  7411
7412  7415  7443  7444  7445  7448  7496  7498  7537  7542  7574  7584  7587
7617  7619  7653  7654  7668  7672  7690  7704  7705  7706  7707  7729  7745
7746  7751  7752  7756  7775  7791  7792  7797  7798  7802  7821  7837  7838
7843  7844  7848  7867  7883  7884  7889  7890  7894  7913  7929  7930  7935
7936  7940  7959  7975  7976  7981  7982  7986  8005  8021  8022  8027  8028
8032  8051  8067  8068  8073  8074  8078  8097  8113  8114  8119  8120  8125
8145  8161  8162  8167  8168  8172  8193  8209  8210  8215  8216  8221  8241
8257  8258  8263  8264  8268  8288  8304  8305  8310  8311  8315  8335  8351
8352  8357  8358  8362  8382  8398  8399  8404  8405  8409  8430  8446  8447
8452  8453  8457  8477  8493  8494  8499  8500  8504  8526  8542  8543  8548
8549  8553  8573  8589  8590  8595  8596  8600  8620  8636  8637  8642  8643
8647  8667  8683  8684  8689  8690  8694  8714  8730  8731  8736  8737  8741
8761  8777  8778  8783  8784  8788  8808  8824  8825  8830  8831  8835  8855
8871  8872  8877  8878  8882  8902  8918  8919  8924  8925  8929  8950  8966
8967  8972  8973  8977  8997  9013  9014  9019  9020  9024  9044  9060  9061
9066  9067  9071  9091  9107  9108  9113  9114  9118  9138  9154  9155  9160
9161  9165  9185  9201  9202  9207  9208  9212  9236  9238  9239  9240  9241
9247  9279  9282  9298
SVCSUB= 000000  4807#  4820#  7106
SVCTAG= 000000  4807#  4822#  4909  4926  5556  5557  5558  5559  5560  5561  5562  5563  5564
5565  5566  5568  5570  5571  5572  5573  5574  5575  5576  5577  5578  5579
5580  5581  5582  5583  5584  5585  5586  5587  5588  5589  5594  5624  5753
5774  5787  5804  5820  5857  5871  5908  5922  5923  5952  5953  5979  5987
5989  6008  6015  6017  6036  6043  6045  6074  6075  6105  6120  6121  6145
6161  6162  6188  6200  6201  6222  6231  6232  6261  6290  6310  6311  6339
6359  6360  6388  6408  6409  6437  6457  6458  6490  6514  6515  6549  6575
6576  6604  6624  6625  6653  6673  6674  6702  6722  6723  6751  6771  6772
6800  6820  6821  6849  6869  6870  6898  6918  6919  6947  6967  6968  7005
7024  7025  7053  7092  7093  7128  7154  7159  7160  7198  7219  7220  7246
7271  7308  7346  7381  7415  7448  7498  7542  7587  7619  7672  7706  7707
7751  7756  7797  7802  7843  7848  7889  7894  7935  7940  7981  7986  8027
8032  8073  8078  8119  8125  8167  8172  8215  8221  8263  8268  8310  8315
8357  8362  8404  8409  8452  8457  8499  8504  8548  8553  8595  8600  8642
8647  8689  8694  8736  8741  8783  8788  8830  8835  8877  8882  8924  8929
8972  8977  9019  9024  9066  9071  9113  9118  9160  9165  9207  9212  9247
9282
SVCTST  000000  4807#  4819#  5838  5863  5891  5931  5968  5997  6025  6052  6089  6129  6170
6209  6245  6269  6319  6368  6417  6466  6523  6584  6633  6682  6731  6780
6829  6878  6927  6976  7033  7101  7168  7227  7253  7279  7316  7353  7388
7422  7456  7510  7551  7595  7627  7683  7720  7766  7812  7858  7904  7950
7996  8042  8088  8136  8183  8232  8279  8326  8373  8421  8468  8517  8564
```

```
                    8611  8658  8705  8752  8799  8846  8893  8941  8988  9035  9082  9128  9176
SWPAC1  035263      9255#
SWPAC2  035324      9256#
S$LSYM= 010000      4807#  4909#  4926#  5556#  5557#  5558#  5559#  5560#  5561#  5562#  5563#  5564#  5565#
                    5566#  5568#  5570#  5571#  5572#  5573#  5574#  5575#  5576#  5577#  5578#  5579#  5580#
                    5581#  5582#  5583#  5584#  5585#  5586#  5587#  5588#  5589#  5594#  5624#  5753#  5774#
                    5787#  5804#  5820#  5857#  5871#  5896#  5911#  5923#  5935#  5953#  5970#  5980#  5989#
                    5999#  6009#  6017#  6027#  6037#  6045#  6056#  6075#  6094#  6107#  6121#  6134#  6147#
                    6162#  6174#  6189#  6201#  6213#  6224#  6232#  6261#  6275#  6293#  6311#  6324#  6342#
                    6360#  6373#  6391#  6409#  6422#  6440#  6458#  6471#  6493#  6515#  6528#  6552#  6576#
                    6589#  6607#  6625#  6638#  6656#  6674#  6687#  6705#  6723#  6736#  6754#  6772#  6785#
                    6803#  6821#  6834#  6852#  6870#  6883#  6901#  6919#  6932#  6950#  6968#  6981#  7008#
                    7025#  7038#  7056#  7093#  7107#  7130#  7159#  7160#  7173#  7201#  7220#  7246#  7271#
                    7308#  7346#  7381#  7415#  7448#  7498#  7542#  7587#  7619#  7672#  7690#  7707#  7729#
                    7756#  7775#  7802#  7821#  7848#  7867#  7894#  7913#  7940#  7959#  7986#  8005#  8032#
                    8051#  8078#  8097#  8125#  8145#  8172#  8193#  8221#  8241#  8268#  8288#  8315#  8335#
                    8362#  8382#  8409#  8430#  8457#  8477#  8504#  8526#  8553#  8573#  8600#  8620#  8647#
                    8667#  8694#  8714#  8741#  8761#  8788#  8808#  8835#  8855#  8882#  8902#  8929#  8950#
                    8977#  8997#  9024#  9044#  9071#  9091#  9118#  9138#  9165#  9185#  9212#  9247#  9282#
TDATA   024470      7687  7689  7708#
TEMP    002444      4994#
TFM1    004107      5445#  5559  5560  5561  5562  5563
TFM2    004132      5446#  5556  5557  5558  5564  5565  5568  5573  5578  5579  5581  5583  5586
                    5587  5588  5589
TFM27   004164      5448#  5580  5582
TFM37   004211      5449#  5593
TFM5    004147      5447#  5570  5576
TOUTP   023534      7536  7540#
TOUTT   023530      7513  7539#
TYPE    002630      5027#  5741*  5746*  5751*
T$ARGC= 000002      4848#  5556#  5557#  5558#  5559#  5560#  5561#  5562#  5563#  5564#  5565#  5566#  5568#
                    5570#  5571#  5572#  5573#  5574#  5575#  5576#  5577#  5578#  5579#  5580#  5581#  5582#
                    5583#  5584#  5585#  5586#  5587#  5588#  5589#  5592#  5593#
T$CODF= 003032      9238#  9239#  9240#  9241#
T$ERRN= 000027      4807#  5383#  5852#  5869#  5902#  5916#  5942#  5948#  5977#  5985#  6006#  6014#  6034#
                    6042#  6054#  6070#  6100#  6114#  6140#  6155#  6180#  6194#  6220#  6229#  6257#  6285#
                    6304#  6334#  6353#  6383#  6402#  6432#  6451#  6484#  6507#  6542#  6568#  6599#  6618#
                    6648#  6667#  6697#  6716#  6746#  6765#  6795#  6814#  6844#  6863#  6893#  6912#  6942#
                    6961#  6999#  7017#  7048#  7068#  7084#  7123#  7148#  7192#  7213#  7241#  7243#  7266#
                    7268#  7305#  7338#  7342#  7376#  7410#  7443#  7496#  7537#  7574#  7584#  7617#  7653#
                    7704#  7745#  7791#  7837#  7883#  7929#  7975#  8021#  8067#  8113#  8161#  8209#  8257#
                    8304#  8351#  8398#  8446#  8493#  8542#  8589#  8636#  8683#  8730#  8777#  8824#  8871#
                    8918#  8966#  9013#  9060#  9107#  9154#  9201#
T$EXCP= 000000      9238#  9239#  9240#  9241#
T$FLAG= 000040      5615#  5845#  5856#  5903#  5917#  5943#  5949#  5978#  5986#  6007#  6035#  6065#  6071#
                    6101#  6115#  6141#  6156#  6181#  6195#  6221#  6230#  6259#  6286#  6305#  6335#  6354#
                    6384#  6403#  6433#  6452#  6485#  6508#  6544#  6569#  6600#  6619#  6649#  6668#  6698#
                    6717#  6747#  6766#  6796#  6815#  6845#  6864#  6894#  6913#  6943#  6962#  7000#  7018#
                    7049#  7069#  7124#  7149#  7193#  7214#  7377#  7378#  7411#  7412#  7444#  7445#  7654#
                    7668#  7705#  7746#  7752#  7792#  7798#  7838#  7844#  7884#  7890#  7930#  7936#  7976#
                    7982#  8022#  8028#  8068#  8074#  8114#  8120#  8162#  8168#  8210#  8216#  8258#  8264#
                    8305#  8311#  8352#  8358#  8399#  8405#  8447#  8453#  8494#  8500#  8543#  8549#  8590#
                    8596#  8637#  8643#  8684#  8690#  8731#  8737#  8778#  8784#  8825#  8831#  8872#  8878#
                    8919#  8925#  8967#  8973#  9014#  9020#  9061#  9067#  9108#  9114#  9155#  9161#  9202#
                    9208#
T$GMAN= 000000      4807#
```

T$HILI= 000007 9238# 9239# 9240# 9241#
T$LAST= 000001 4807# 9298#
T$LOLI= 000004 9238# 9239# 9240# 9241#
T$LSYM= 010000 4807# 4909 4926 5556 5557 5558 5559 5560 5561 5562 5563 5564 5565
5566 5568 5570 5571 5572 5573 5574 5575 5576 5577 5578 5579 5580
5581 5582 5583 5584 5585 5586 5587 5588 5589 5594 5624 5753 5774
5787 5804 5820 5857 5871 5923 5953 5989 6017 6045 6075 6121 6162
6201 6232 6261 6311 6360 6409 6458 6515 6576 6625 6674 6723 6772
6821 6870 6919 6968 7025 7093 7159 7160 7220 7246 7271 7308 7346
7381 7415 7448 7498 7542 7587 7619 7672 7707 7756 7802 7848 7894
7940 7986 8032 8078 8125 8172 8221 8268 8315 8362 8409 8457 8504
8553 8600 8647 8694 8741 8788 8835 8882 8929 8977 9024 9071 9118
9165 9212 9247 9282
T$LTNO= 000114 9298#
T$NEST= 177777 4807# 4813# 4857# 4861# 4894# 4909# 4923# 4926# 5556# 5557# 5558# 5559# 5560#
5561# 5562# 5563# 5564# 5565# 5566# 5568# 5570# 5571# 5572# 5573# 5574# 5575#
5576# 5577# 5578# 5579# 5580# 5581# 5582# 5583# 5584# 5585# 5586# 5587# 5588#
5589# 5591# 5594# 5607# 5624# 5638# 5753# 5756# 5774# 5784# 5787# 5801# 5804#
5819# 5820# 5838# 5857# 5863# 5871# 5891# 5896# 5908# 5911# 5922# 5923# 5931#
5935# 5952# 5953# 5968# 5970# 5979# 5980# 5987# 5989# 5997# 5999# 6008# 6009#
6015# 6017# 6025# 6027# 6036# 6037# 6043# 6045# 6052# 6056# 6074# 6075# 6089#
6094# 6105# 6107# 6120# 6121# 6129# 6134# 6145# 6147# 6161# 6162# 6170# 6174#
6188# 6189# 6200# 6201# 6209# 6213# 6222# 6224# 6231# 6232# 6245# 6261# 6269#
6275# 6290# 6293# 6310# 6311# 6319# 6324# 6339# 6342# 6359# 6360# 6368# 6373#
6398# 6391# 6408# 6409# 6417# 6422# 6437# 6440# 6457# 6458# 6466# 6471# 6490#
6493# 6514# 6515# 6523# 6528# 6549# 6552# 6575# 6576# 6584# 6593# 6604# 6607#
6624# 6625# 6633# 6638# 6653# 6656# 6673# 6674# 6682# 6687# 6702# 6705# 6722#
6723# 6731# 6736# 6751# 6754# 6771# 6772# 6780# 6785# 6800# 6803# 6820# 6821#
6829# 6834# 6849# 6852# 6869# 6870# 6878# 6883# 6898# 6901# 6918# 6919# 6927#
6932# 6947# 6950# 6967# 6968# 6976# 6981# 7005# 7008# 7024# 7025# 7033# 7038#
7053# 7056# 7092# 7093# 7101# 7106# 7107# 7128# 7130# 7154# 7159# 7160# 7168#
7173# 7198# 7201# 7219# 7220# 7227# 7246# 7253# 7271# 7279# 7308# 7316# 7346#
7353# 7381# 7388# 7415# 7422# 7448# 7456# 7498# 7510# 7542# 7551# 7587# 7595#
7619# 7627# 7672# 7683# 7690# 7706# 7707# 7720# 7729# 7751# 7756# 7766# 7775#
7797# 7802# 7812# 7821# 7843# 7848# 7858# 7867# 7889# 7894# 7904# 7913# 7935#
7940# 7950# 7959# 7981# 7986# 7996# 8005# 8027# 8032# 8042# 8051# 8073# 8078#
8088# 8097# 8119# 8125# 8136# 8145# 8167# 8172# 8183# 8193# 8215# 8221# 8232#
8241# 8263# 8268# 8279# 8288# 8310# 8315# 8326# 8335# 8357# 8362# 8373# 8382#
8404# 8409# 8421# 8430# 8452# 8457# 8468# 8477# 8499# 8504# 8517# 8526# 8548#
8553# 8564# 8573# 8595# 8600# 8611# 8620# 8642# 8647# 8658# 8667# 8689# 8694#
8705# 8714# 8736# 8741# 8752# 8761# 8783# 8788# 8799# 8808# 8830# 8835# 8846#
8855# 8877# 8882# 8893# 8902# 8924# 8929# 8941# 8950# 8972# 8977# 8988# 8997#
9019# 9024# 9035# 9044# 9066# 9071# 9082# 9091# 9113# 9118# 9128# 9138# 9160#
9165# 9176# 9185# 9207# 9212# 9236# 9247# 9279# 9282# 9290#
T$NS0 = 000000 4813# 9290
T$NS1 = 000005 4857# 4861 4894# 4909 4923# 4926 5556# 5557# 5558# 5559# 5560# 5561# 5562#
5563# 5564# 5565# 5566# 5568# 5570# 5571# 5572# 5573# 5574# 5575# 5576# 5577#
5578# 5579# 5580# 5581# 5582# 5583# 5584# 5585# 5586# 5587# 5588# 5589# 5591#
5594 5607# 5624 5638# 5753 5756# 5774 5784# 5787 5801# 5804 5819# 5820
5838# 5857 5863# 5871 5891# 5923 5931# 5953 5968# 5989 5997# 6017 6025#
6045 6052# 6075 6089# 6121 6129# 6162 6170# 6201 6209# 6232 6245# 6261
6269# 6311 6319# 6360 6368# 6409 6417# 6458 6466# 6515 6523# 6576 6584#
6625 6633# 6674 6682# 6723 6731# 6772 6780# 6821 6829# 6870 6878# 6919
6927# 6968 6976# 7025 7033# 7093 7101# 7160 7168# 7220 7227# 7246 7253#
7271 7279# 7308 7316# 7346 7353# 7381 7388# 7415 7422# 7448 7456# 7498
7510# 7542 7551# 7587 7595# 7619 7627# 7672 7683# 7707 7720# 7756 7766#

```
                    7802    7812#   7848    7858#   7894    7904#   7940    7950#   7986    7996#   8032    8042#   8078
                    8088#   8125    8136#   8172    8183#   8221    8232#   8268    8279#   8315    8326#   8362    8373#
                    8409    8421#   8457    8468#   8504    8517#   8553    8564#   8600    8611#   8647    8658#   8694
                    8705#   8741    8752#   8788    8799#   8835    8846#   8882    8893#   8929    8941#   8977    8988#
                    9024    9035#   9071    9082#   9118    9128#   9165    9176#   9212    9236#   9247    9279#   9282
I$NS2 = 000003      5896#   5908    5911#   5922    5935#   5952    5970#   5979    5980#   5987    5999#   6008    6009#
                    6015    6027#   6036    6037#   6043    6056#   6074    6094#   6105    6107#   6120    6134#   6145
                    6147#   6161    6174#   6188    6189#   6200    6213#   6222    6224#   6231    6275#   6290    6293#
                    6310    6324#   6339    6342#   6359    6373#   6388    6391#   6408    6422#   6437    6440#   6457
                    6471#   6490    6493#   6514    6528#   6549    6552#   6575    6589#   6604    6607#   6624    6638#
                    6653    6656#   6673    6687#   6702    6705#   6722    6736#   6751    6754#   6771    6785#   6800
                    6803#   6820    6834#   6849    6852#   6869    6883#   6898    6901#   6918    6932#   6947    6950#
                    6967    6981#   7005    7008#   7024    7038#   7053    7056#   7092    7106#   7159    7173#   7198
                    7201#   7219    7690#   7706    7729#   7751    7775#   7797    7821#   7843    7867#   7889    7913#
                    7935    7959#   7981    8005#   8027    8051#   8073    8097#   8119    8145#   8167    8193#   8215
                    8241#   8263    8288#   8310    8335#   8357    8382#   8404    8430#   8452    8477#   8499    8526#
                    8548    8573#   8595    8620#   8642    8667#   8689    8714#   8736    8761#   8783    8808#   8830
                    8855#   8877    8902#   8924    8950#   8972    8997#   9019    9044#   9066    9091#   9113    9138#
                    9160    9185#   9207
T$NS3 = 000003      7107#   7128    7130#   7154
T$PTNU= 000000      4807#
T$SAVL= 177777      4807#
T$SEGL= 177777      4807#   5896#   5903    5908#   5911#   5917    5922#   5935#   5943    5949    5952#   5970#   5978
                    5979#   5980#   5986    5987#   5999#   6007    6008#   6009#   6015#   6027#   6035    6036#   6037#
                    6043#   6056#   6065    6071    6074#   6094#   6101    6105#   6107#   6115    6120#   6134#   6141
                    6145#   6147#   6156    6161#   6174#   6181    6188#   6189#   6195    6200#   6213#   6221    6222#
                    6224#   6230    6231#   6275#   6286    6290#   6293#   6305    6310#   6324#   6335    6339#   6342#
                    6354    6359#   6373#   6384    6388#   6391#   6403    6408#   6422#   6433    6437#   6440#   6452
                    6457#   6471#   6485    6490#   6493#   6508    6514#   6528#   6544    6549#   6552#   6569    6575#
                    6589#   6600    6604#   6607#   6619    6624#   6638#   6649    6653#   6656#   6668    6673#   6687#
                    6698    6702#   6705#   6717    6722#   6736#   6747    6751#   6754#   6766    6771#   6785#   6796
                    6800#   6803#   6815    6820#   6834#   6845    6849#   6852#   6864    6869#   6883#   6894    6898#
                    6901#   6913    6918#   6932#   6943    6947#   6950#   6962    6967#   6981#   7000    7005#   7008#
                    7018    7024#   7038#   7049    7053#   7056#   7069    7092#   7107#   7124    7128#   7130#   7154#
                    7173#   7193    7198#   7201#   7214    7219#   7690#   7705    7706#   7729#   7746    7751#   7775#
                    7792    7797#   7821#   7838    7843#   7867#   7884    7889#   7913#   7930    7935#   7959#   7976
                    7981#   8005#   8022    8027#   8051#   8068    8073#   8097#   8114    8119#   8145#   8162    8167#
                    8193#   8210    8215#   8241#   8258    8263#   8288#   8305    8310#   8335#   8352    8357#   8382#
                    8399    8404#   8430#   8447    8452#   8477#   8494    8499#   8526#   8543    8548#   8573#   8590
                    8595#   8620#   8637    8642#   8667#   8684    8689#   8714#   8731    8736#   8761#   8778    8783#
                    8808#   8825    8830#   8855#   8872    8877#   8902#   8919    8924#   8950#   8967    8972#   8997#
                    9014    9019#   9044#   9061    9066#   9091#   9108    9113#   9138#   9155    9160#   9185#   9202
                    9207#
T$SEK0= 010000      5896#   5903    5908    5911#   5917    5922    5935#   5943    5949    5952    5970#   5978    5979
                    5980#   5986    5987    5999#   6007    6008    6009#   6015    6027#   6035    6036    6037#   6043
                    6056#   6065    6071    6074    6094#   6101    6105    6107#   6115    6120    6134#   6141    6145
                    6147#   6156    6161    6174#   6181    6188    6189#   6195    6200    6213#   6221    6222    6224#
                    6230    6231    6275#   6286    6290    6293#   6305    6310    6324#   6335    6339    6342#   6354
                    6359    6373#   6384    6388    6391#   6403    6408    6422#   6433    6437    6440#   6452    6457
                    6471#   6485    6490    6493#   6508    6514    6528#   6544    6549    6552#   6569    6575    6589#
                    6600    6604    6607#   6619    6624    6638#   6649    6653    6656#   6668    6673    6687#   6698
                    6702    6705#   6717    6722    6736#   6747    6751    6754#   6766    6771    6785#   6796    6800
                    6803#   6815    6820    6834#   6845    6849    6852#   6864    6869    6883#   6894    6898    6901#
                    6913    6918    6932#   6943    6947    6950#   6962    6967    6981#   7000    7005    7008#   7018
                    7024    7038#   7049    7053    7056#   7069    7092    7107#   7124    7128    7130#   7154    7173#
                    7193    7198    7201#   7214    7219    7690#   7705    7706    7729#   7746    7751    7775#   7792
```

N 1

T$SUBN= 000000
T$TAGL= 177777
T$TAGN= 010171
T$TEMP= 000000
T$TEST= 000114

7797 7821# 7838 7843 7867# 7884 7889 7913# 7930 7935 7959# 7976 7981
8005# 8022 8027 8051# 8068 8073 8097# 8114 8119 8145# 8162 8167 8193#
8210 8215 8241# 8258 8263 8288# 8305 8310 8335# 8352 8357 8382# 8399
8404 8430# 8447 8452 8477# 8494 8499 8526# 8543 8548 8573# 8590 8595
8620# 8637 8642 8667# 8684 8689 8714# 8731 8736 8761# 8778 8783 8808#
8825 8830 8855# 8872 8877 8902# 8919 8924 8950# 8967 8972 8997# 9014
9019 9044# 9061 9066 9091# 9108 9113 9138# 9155 9160 9185# 9202 9207
4807# 5838# 5863# 5891# 5931# 5968# 5997# 6025# 6052# 6089# 6129# 6170# 6209#
6245# 6269# 6319# 6368# 6417# 6466# 6523# 6584# 6633# 6682# 6731# 6780# 6829#
6878# 6927# 6976# 7033# 7101# 7106# 7168# 7227# 7253# 7279# 7316# 7353# 7388#
7422# 7456# 7510# 7551# 7595# 7627# 7683# 7720# 7766# 7812# 7858# 7904# 7950#
7996# 8042# 8088# 8136# 8183# 8232# 8279# 8326# 8373# 8421# 8468# 8517# 8564#
8611# 8658# 8705# 8752# 8799# 8846# 8893# 8941# 8988# 9035# 9082# 9128# 9176#
4807#
4807# 4857# 4894# 4923# 5556# 5557# 5558# 5559# 5560# 5561# 5562# 5563# 5564#
5565# 5566# 5568# 5570# 5571# 5572# 5573# 5574# 5575# 5576# 5577# 5578# 5579#
5580# 5581# 5582# 5583# 5584# 5585# 5586# 5587# 5588# 5589# 5591# 5607# 5638#
5756# 5784# 5801# 5819# 5838# 5863# 5891# 5931# 5968# 5997# 6025# 6052# 6089#
6129# 6170# 6209# 6245# 6269# 6319# 6368# 6417# 6466# 6523# 6584# 6633# 6682#
6731# 6780# 6829# 6878# 6927# 6976# 7033# 7101# 7106# 7168# 7227# 7253# 7279#
7316# 7353# 7388# 7422# 7456# 7510# 7551# 7595# 7627# 7683# 7720# 7766# 7812#
7858# 7904# 7950# 7996# 8042# 8088# 8136# 8183# 8232# 8279# 8326# 8373# 8421#
8468# 8517# 8564# 8611# 8658# 8705# 8752# 8799# 8846# 8893# 8941# 8988# 9035#
9082# 9128# 9176# 9236# 9279#
4861# 4871# 4909# 4926# 5556# 5557# 5558# 5559# 5560# 5561# 5562# 5563# 5564#
5565# 5566# 5568# 5570# 5571# 5572# 5573# 5574# 5575# 5576# 5577# 5578# 5579#
5580# 5581# 5582# 5583# 5584# 5585# 5586# 5587# 5588# 5589# 5594# 5615# 5624#
5753# 5774# 5787# 5804# 5820# 5845# 5856# 5857# 5871# 5903# 5908# 5917# 5922#
5923# 5943# 5949# 5952# 5953# 5978# 5979# 5986# 5987# 5989# 6007# 6008# 6015#
6017# 6035# 6036# 6043# 6045# 6065# 6071# 6074# 6075# 6101# 6105# 6115# 6120#
6121# 6141# 6145# 6156# 6161# 6162# 6181# 6188# 6195# 6200# 6201# 6221# 6222#
6230# 6231# 6232# 6259# 6261# 6286# 6290# 6305# 6310# 6311# 6335# 6339# 6354#
6359# 6360# 6384# 6388# 6403# 6408# 6409# 6433# 6437# 6452# 6457# 6458# 6485#
6490# 6508# 6514# 6515# 6544# 6549# 6569# 6575# 6576# 6600# 6604# 6619# 6624#
6625# 6649# 6653# 6668# 6673# 6674# 6698# 6702# 6717# 6722# 6723# 6747# 6751#
6766# 6771# 6772# 6796# 6800# 6815# 6820# 6821# 6845# 6849# 6864# 6869# 6870#
6894# 6898# 6913# 6918# 6919# 6943# 6947# 6962# 6967# 6968# 7000# 7005# 7018#
7024# 7025# 7049# 7053# 7069# 7092# 7093# 7124# 7128# 7149# 7154# 7159# 7160#
7193# 7198# 7214# 7219# 7220# 7246# 7271# 7308# 7346# 7377# 7378# 7381# 7411#
7412# 7415# 7444# 7445# 7448# 7498# 7542# 7587# 7619# 7654# 7668# 7672# 7705#
7706# 7707# 7746# 7751# 7752# 7756# 7792# 7797# 7798# 7802# 7838# 7843# 7844#
7848# 7884# 7889# 7890# 7894# 7930# 7935# 7936# 7940# 7976# 7981# 7982# 7986#
8022# 8027# 8028# 8032# 8068# 8073# 8074# 8078# 8114# 8119# 8120# 8125# 8162#
8167# 8168# 8172# 8210# 8215# 8216# 8221# 8258# 8263# 8264# 8268# 8305# 8310#
8311# 8315# 8352# 8357# 8358# 8362# 8399# 8404# 8405# 8409# 8447# 8452# 8453#
8457# 8494# 8499# 8500# 8504# 8543# 8548# 8549# 8553# 8590# 8595# 8596# 8600#
8637# 8642# 8643# 8647# 8684# 8689# 8690# 8694# 8731# 8736# 8737# 8741# 8778#
8783# 8784# 8788# 8825# 8830# 8831# 8835# 8872# 8877# 8878# 8882# 8919# 8924#
8925# 8929# 8967# 8972# 8973# 8977# 9014# 9019# 9020# 9024# 9061# 9066# 9067#
9071# 9108# 9113# 9114# 9118# 9155# 9160# 9161# 9165# 9202# 9207# 9208# 9212#
9238# 9239# 9240# 9241# 9247# 9282# 9290#
4807# 5833 5836 5838# 5859 5861 5863# 5873 5889 5891# 5925 5929 5931#
5962 5966 5968# 5991 5995 5997# 6019 6023 6025# 6047 6050 6052# 6082
6086 6089# 6123 6127 6129# 6164 6168 6170# 6203 6207 6209# 6237 6243
6245# 6263 6267 6269# 6313 6317 6319# 6362 6366 6368# 6411 6415 6417#
6460 6464 6466# 6517 6521 6523# 6578 6582 6584# 6627 6631 6633# 6676

```
                    6680    6682#   6725    6729    6731#   6774    6778    6780#   6823    6827    6829#   6872    6876
                    6878#   6921    6925    6927#   6970    6974    6976#   7027    7031    7033#   7095    7099    7101#
                    7106    7162    7166    7168#   7222    7225    7227#   7248    7251    7253#   7273    7277    7279#
                    7310    7314    7316#   7348    7351    7353#   7383    7386    7388#   7417    7420    7422#   7450
                    7454    7456#   7500    7508    7510#   7544    7549    7551#   7589    7593    7595#   7621    7625
                    7627#   7678    7681    7683#   7712    7718    7720#   7758    7764    7766#   7804    7810    7812#
                    7850    7856    7858#   7896    7902    7904#   7942    7948    7950#   7988    7994    7996#   8034
                    8040    8042#   8080    8086    8088#   8128    8134    8136#   8175    8181    8183#   8224    8230
                    8232#   8271    8277    8279#   8318    8324    8326#   8365    8371    8373#   8412    8419    8421#
                    8460    8466    8468#   8507    8513    8517#   8556    8562    8564#   8603    8609    8611#   8650
                    8656    8658#   8697    8704    8705#   8744    8750    8752#   8791    8797    8799#   8838    8844
                    8846#   8885    8891    8893#   8932    8938    8941#   8980    8986    8988#   9027    9033    9035#
                    9074    9080    9082#   9120    9126    9128#   9168    9174    9176#   9298    9299
T$TSTM= 177777      4807#   5383    5556    5557    5558    5559    5560    5561    5562    5563    5564    5565    5566
                    5568    5570    5571    5572    5573    5574    5575    5576    5577    5578    5579    5580    5581
                    5582    5583    5584    5585    5586    5587    5588    5589    5592    5593    5594    5624    5652
                    5655    5658    5662    5675    5753    5770    5774    5785    5787    5803    5804    5820    5845
                    5852    5856    5857    5869    5871    5896    5902    5903    5908    5911    5916    5917    5922
                    5923    5935    5942    5943    5948    5949    5952    5953    5970    5977    5978    5979    5980
                    5985    5986    5987    5989    5999    6006    6007    6008    6009    6014    6015    6017    6027
                    6034    6035    6036    6037    6042    6043    6045    6056    6064    6065    6070    6071    6074
                    6075    6094    6100    6101    6105    6107    6114    6115    6120    6121    6134    6140    6141
                    6145    6147    6155    6156    6161    6162    6174    6180    6181    6188    6189    6194    6195
                    6200    6201    6213    6220    6221    6222    6224    6229    6230    6231    6232    6257    6259
                    6261    6275    6285    6286    6290    6293    6304    6305    6310    6311    6324    6334    6335
                    6339    6342    6353    6354    6359    6360    6373    6383    6384    6388    6391    6402    6403
                    6408    6409    6422    6432    6433    6437    6440    6451    6452    6457    6458    6471    6484
                    6485    6490    6493    6507    6508    6514    6515    6528    6542    6544    6549    6552    6568
                    6569    6575    6576    6589    6599    6600    6604    6607    6618    6619    6624    6625    6638
                    6648    6649    6653    6656    6667    6668    6673    6674    6687    6697    6698    6702    6705
                    6716    6717    6722    6723    6736    6746    6747    6751    6754    6765    6766    6771    6772
                    6785    6795    6796    6800    6803    6814    6815    6820    6821    6834    6844    6845    6849
                    6852    6863    6864    6869    6870    6883    6893    6894    6898    6901    6912    6913    6918
                    6919    6932    6942    6943    6947    6950    6961    6962    6967    6968    6981    6999    7000
                    7005    7008    7017    7018    7024    7025    7038    7048    7049    7053    7056    7068    7069
                    7084    7085    7092    7093    7106    7107    7123    7124    7128    7130    7148    7149    7154
                    7159    7160    7173    7192    7193    7198    7201    7213    7214    7219    7220    7229    7235
                    7239    7241    7243    7246    7260    7264    7266    7268    7271    7283    7298    7304    7305
                    7308    7320    7336    7338    7342    7346    7354    7376    7377    7378    7381    7410    7411
                    7412    7415    7443    7444    7445    7448    7496    7498    7537    7542    7574    7584    7587
                    7617    7619    7653    7654    7668    7672    7690    7704    7705    7706    7707    7729    7745
                    7746    7751    7752    7756    7775    7791    7792    7797    7798    7802    7821    7837    7838
                    7843    7844    7848    7867    7883    7884    7889    7890    7894    7913    7929    7930    7935
                    7936    7940    7959    7975    7976    7981    7982    7986    8005    8021    8022    8027    8028
                    8032    8051    8067    8068    8073    8074    8078    8097    8113    8114    8119    8120    8125
                    8145    8161    8162    8167    8168    8172    8193    8209    8210    8215    8216    8221    8241
                    8257    8258    8263    8264    8268    8288    8304    8305    8310    8311    8315    8335    8351
                    8352    8357    8358    8362    8382    8398    8399    8404    8405    8409    8430    8446    8447
                    8452    8453    8457    8477    8493    8494    8499    8500    8504    8526    8542    8543    8548
                    8549    8553    8573    8589    8590    8595    8596    8600    8620    8636    8637    8642    8643
                    8647    8667    8683    8684    8689    8690    8694    8714    8730    8731    8736    8737    8741
                    8761    8777    8778    8783    8784    8788    8808    8824    8825    8830    8831    8835    8855
                    8871    8872    8877    8878    8882    8902    8918    8919    8924    8925    8929    8950    8966
                    8967    8972    8973    8977    8997    9013    9014    9019    9020    9024    9044    9060    9061
                    9066    9067    9071    9091    9107    9108    9113    9114    9118    9138    9154    9155    9160
                    9161    9165    9185    9201    9202    9207    9208    9212
```

```
T$TSTS= 000001          4807#  5838#  5863#  5891#  5931#  5968#  5997#  6025#  6052#  6089#  6129#  6170#  6209#
                        6245#  6269#  6319#  6368#  6417#  6466#  6523#  6584#  6633#  6682#  6731#  6780#  6829#
                        6878#  6927#  6976#  7033#  7101#  7168#  7227#  7253#  7279#  7316#  7353#  7388#  7422#
                        7456#  7510#  7551#  7595#  7627#  7683#  7720#  7766#  7812#  7858#  7904#  7950#  7996#
                        8042#  8088#  8136#  8183#  8232#  8279#  8326#  8373#  8421#  8468#  8517#  8564#  8611#
                        8658#  8705#  8752#  8799#  8846#  8893#  8941#  8988#  9035#  9082#  9128#  9176#
T$$AU = 010051          5819#  5820
T$$AUT= 010046          5756#  5774
T$$CLE= 010047          5784#  5787
T$$DU = 010050          5801#  5804
T$$HAR= 010167          9236#  9247
T$$HW = 010001          4894#  4909
T$$INI= 010045          5638#  5753
T$$MSG= 010043          5556#  5557#  5558#  5559#  5560#  5561#  5562#  5563#  5564#  5565#  5566#  5568#  5570#
                        5571#  5572#  5573#  5574#  5575#  5576#  5577#  5578#  5579#  5580#  5581#  5582#  5583#
                        5584#  5585#  5586#  5587#  5588#  5589#  5591#  5594
T$$PRO= 010000          4857#
T$$RPT= 010044          5607#  5615  5624
T$$SEG= 010000          5896#  5903  5908#  5911#  5917  5922#  5935#  5943  5949  5952#  5970#  5978  5979#
                        5980#  5986  5987#  5999#  6007  6008#  6009#  6015#  6027#  6035  6036#  6037#  6043#
                        6056#  6065  6071  6074#  6094#  6101  6105#  6107#  6115  6120#  6134#  6141  6145#
                        6147#  6156  6161#  6174#  6181  6188#  6189#  6195  6200#  6213#  6221  6222#  6224#
                        6230  6231#  6275#  6286  6290#  6293#  6305  6310#  6324#  6335  6339#  6342#  6354
                        6359#  6373#  6384  6388#  6391#  6403  6408#  6422#  6433  6437#  6440#  6452  6457#
                        6471#  6485  6490#  6493#  6508  6514#  6528#  6544  6549#  6552#  6569  6575#  6589#
                        6600  6604#  6607#  6619  6624#  6638#  6649  6653#  6656#  6668  6673#  6687#  6698
                        6702#  6705#  6717  6722#  6736#  6747  6751#  6754#  6766  6771#  6785#  6796  6800#
                        6803#  6815  6820#  6834#  6845  6849#  6852#  6864  6869#  6883#  6894  6898#  6901#
                        6913  6918#  6932#  6943  6947#  6950#  6962  6967#  6981#  7000  7005#  7008#  7018
                        7024#  7038#  7049  7053#  7056#  7069  7092#  7107#  7124  7128#  7130#  7154#  7173#
                        7193  7198#  7201#  7214  7219#  7690#  7705  7706#  7729#  7746  7751#  7775#  7792
                        7797#  7821#  7838  7843#  7867#  7884  7889#  7913#  7930  7935#  7959#  7976  7981#
                        8005#  8022  8027#  8051#  8068  8073#  8097#  8114  8119#  8145#  8162  8167#  8193#
                        8210  8215#  8241#  8258  8263#  8288#  8305  8310#  8335#  8352  8357#  8382#  8399
                        8404#  8430#  8447  8452#  8477#  8494  8499#  8526#  8543  8548#  8573#  8590  8595#
                        8620#  8637  8642#  8667#  8684  8689#  8714#  8731  8736#  8761#  8778  8783#  8808#
                        8825  8830#  8855#  8872  8877#  8902#  8919  8924#  8950#  8967  8972#  8997#  9014
                        9019#  9044#  9061  9066#  9091#  9108  9113#  9138#  9155  9160#  9185#  9202  9207#
T$$SOF= 010170          9279#  9282
T$$SUB= 010110          7106#  7159
T$$SW = 010002          4923#  4926
T$$TES= 010166          5838#  5845  5856  5857  5863#  5871  5891#  5923  5931#  5953  5968#  5989  5997#
                        6017  6025#  6045  6052#  6075  6089#  6121  6129#  6162  6170#  6201  6209#  6232
                        6245#  6259  6261  6269#  6311  6319#  6360  6368#  6409  6417#  6458  6466#  6515
                        6523#  6576  6584#  6625  6633#  6674  6682#  6723  6731#  6772  6780#  6821  6829#
                        6870  6878#  6919  6927#  6968  6976#  7025  7033#  7093  7101#  7149  7160  7168#
                        7220  7227#  7246  7253#  7271  7279#  7308  7316#  7346  7353#  7377  7378  7381
                        7388#  7411  7412  7415  7422#  7444  7445  7448  7456#  7498  7510#  7542  7551#
                        7587  7595#  7619  7627#  7654  7668  7672  7683#  7707  7720#  7752  7756  7766#
                        7798  7802  7812#  7844  7848  7858#  7890  7894  7904#  7936  7940  7950#  7982
                        7986  7996#  8028  8032  8042#  8074  8078  8088#  8120  8125  8136#  8168  8172
                        8183#  8216  8221  8232#  8264  8268  8279#  8311  8315  8326#  8358  8362  8373#
                        8405  8409  8421#  8453  8457  8468#  8500  8504  8517#  8549  8553  8564#  8596
                        8600  8611#  8643  8647  8658#  8690  8694  8705#  8737  8741  8752#  8784  8788
                        8799#  8831  8835  8846#  8878  8882  8893#  8925  8929  8941#  8973  8977  8988#
                        9020  9024  9035#  9067  9071  9082#  9114  9118  9128#  9161  9165  9176#  9208
```

9212

T1	011352	G	4871	5838#
T10	013040	G	4871	6129#
T11	013224	G	4871	6170#
T12	013412	G	4871	6209#
T13	013562	G	4871	6245#
T14	013670	G	4871	6269#
T15	014120	G	4871	6319#
T16	014350	G	4871	6368#
T17	014600	G	4871	6417#
T18	015030	G	4871	6466#
T19	015314	G	4871	6523#
T2	011500	G	4871	5863#
T20	015610	G	4871	6584#
T21	016040	G	4871	6633#
T22	016270	G	4871	6682#
T23	016520	G	4871	6731#
T24	016750	G	4871	6780#
T25	017200	G	4871	6829#
T26	017430	G	4871	6878#
T27	017660	G	4871	6927#
T28	020110	G	4871	6976#
T29	020406	G	4871	7033#
T3	011544	G	4871	5891#
T30	020726	G	4871	7101#
T30.1	020744		7106#	
T31	021272	G	4871	7168#
T32	021604	G	4871	7227#
T33	021746	G	4871	7253#
T34	022110	G	4871	7279#
T35	022264	G	4871	7316#
T36	022470	G	4871	7353#
T37	022634	G	4871	7388#
T38	023004	G	4871	7422#
T39	023150	G	4871	7456#
T4	011732	G	4871	5931#
T40	023336	G	4871	7510#
T41	023552	G	4871	7551#
T42	023770	G	4871	7595#
T43	024122	G	4871	7627#
T44	024340	G	4871	7683#
T45	024500	G	4871	7720#
T46	024704	G	4871	7766#
T47	025110	G	4871	7812#
T48	025314	G	4871	7858#
T49	025520	G	4871	7904#
T5	012076	G	4871	5968#
T50	025724	G	4871	7950#
T51	026130	G	4871	7996#
T52	026334	G	4871	8042#
T53	026540	G	4871	8088#
T54	026746	G	4871	8136#
T55	027152	G	4871	8183#
T56	027356	G	4871	8232#
T57	027562	G	4871	8279#
T58	027766	G	4871	8326#

```
T59      030172 G          4871    8373#
T6       012232 G          4871    5997#
T60      030376 G          4871    8421#
T61      030602 G          4871    8468#
T62      031006 G          4871    8517#
T63      031212 G          4871    8564#
T64      031416 G          4871    8611#
T65      031622 G          4871    8658#
T66      032026 G          4871    8705#
T67      032232 G          4871    8752#
T68      032436 G          4871    8799#
T69      032642 G          4871    8846#
T7       012362 G          4871    6025#
T70      033046 G          4871    8893#
T71      033252 G          4871    8941#
T72      033456 G          4871    8988#
T73      033662 G          4871    9035#
T74      034066 G          4871    9082#
T75      034272 G          4871    9128#
T76      034476 G          4871    9176#
T8       012512 G          4871    6052#
T9       012654 G          4871    6089#
UAM    - 000200 G          4953#
VECTOR   035066            9240    9252#
WMP      034750            9238    9249#
WTYPE    002626            5026#   5678*   5742    5744    5749
X$ALWA=  000000            4807#
X$FALS=  000040            4807#
X$OFFS=  000400            4807#
X$TRUE=  000020            4807#
ZERO     002600            5014#
$BDADR   002634            5029#   5383*   5852*   5869*   5902*   5916*   5942*   5948*   5977*   5985*   6006*   6014*   6034*
                           6042*   6064*   6070*   6100*   6114*   6140*   6155*   6180*   6194*   6220*   6229*   6257*   7241*
                           7243*   7266*   7268*   7305*   7338*   7342*   7376*   7410*   7443*   7496*   7537*   7574*   7584*
                           7617*   7653*   7704*   7745*   7791*   7837*   7883*   7929*   7975*   8021*   8067*   8113*   8161*
                           8209*   8257*   8304*   8351*   8398*   8446*   8493*   8542*   8589*   8636*   8683*   8730*   8777*
                           8824*   8871*   8918*   8966*   9013*   9060*   9107*   9154*   9201*
$BDDAT   002640            5031#   5383*   5556    5557    5558    5559    5560    5561    5562    5563    5564    5565    5568
                           5570    5573    5576    5578    5579    5580    5581    5582    5583    5586    5587    5588    5589
                           5852*   5869*   5902*   5916*   5942*   5948*   5977*   5985*   6006*   6014*   6034*   6042*   6064*
                           6070*   6100*   6114*   6140*   6155*   6180*   6194*   6220*   6229*   6257*   6285*   6304*   6334*
                           6353*   6383*   6402*   6432*   6451*   6484*   6507*   6542*   6568*   6599*   6618*   6648*   6667*
                           6697*   6716*   6746*   6765*   6795*   6814*   6844*   6863*   6893*   6912*   6942*   6961*   6999*
                           7017*   7048*   7068*   7084*   7123*   7148*   7192*   7213*   7241*   7243*   7266*   7268*   7305*
                           7338*   7342*   7376*   7410*   7443*   7496*   7537*   7574*   7584*   7617*   7653*   7704*   7745*
                           7791*   7837*   7883*   7929*   7975*   8021*   8067*   8113*   8161*   8209*   8257*   8304*   8351*
                           8398*   8446*   8493*   8542*   8589*   8636*   8683*   8730*   8777*   8824*   8871*   8918*   8966*
                           9013*   9060*   9107*   9154*   9201*
$GDADR   002632            5028#   5559    5560    5561    5562    5563    5593    5769*   5851*   7123*   7148*   7192*   7213*
$GDDAT   002636            5030#   5379*   5381    5383*   5556    5557    5558    5559    5560    5561    5562    5563    5564
                           5565    5568    5570    5573    5576    5578    5579    5580    5581    5582    5583    5586    5587
                           5588    5589    5852*   5869*   5902*   5916*   5937*   5938    5940    5945*   5948*   5977*   5985*
                           6006*   6014*   6034*   6042*   6059*   6060    6062    6067*   6100*   6114*   6140*   6155*   6180*
                           6194*   6220*   6229*   6257*   6285*   6304*   6334*   6353*   6383*   6402*   6432*   6451*   6484*
                           6507*   6542*   6568*   6599*   6618*   6648*   6667*   6697*   6716*   6746*   6765*   6795*   6814*
                           6844*   6863*   6893*   6912*   6942*   6961*   6999*   7017*   7048*   7068*   7084*   7119*   7121
```

```
                    7123* 7144* 7146  7148* 7188* 7190  7192* 7209* 7211  7213* 7241* 7243* 7266*
                    7268* 7290* 7305* 7338* 7342* 7372* 7374  7402* 7408  7439* 7441  7474* 7494
                    7533* 7534* 7535  7569* 7572  7580* 7612* 7615  7649* 7651  7700* 7702  7741*
                    7743  7787* 7789  7833* 7835  7879* 7881  7925* 7927  7971* 7973  8017* 8019
                    8063* 8065  8109* 8111  8157* 8159  8205* 8207  8253* 8255  8300* 8302  8347*
                    8349  8394* 8396  8442* 8444  8489* 8491  8538* 8540  8585* 8587  8632* 8634
                    8679* 8681  8726* 8728  8773* 8775  8820* 8822  8867* 8869  8914* 8916  8962*
                    8964  9009* 9011  9056* 9058  9103* 9105  9150* 9152  9197* 9199
$LSTIN= 000000      4816#
$LSTTA= 000000      4817#
$TMPO   002550      5002# 5357* 5360  5375
.       040004      4799# 4995# 4997# 5019# 5020# 5021# 5022# 5049# 5095# 5212# 5615  5845  5856
                    5903  5917  5943  5949  5978  5986  6007  6035  6065  6071  6101  6115  6141
                    6156  6181  6195  6221  6230  6259  6286  6305  6335  6354  6384  6403  6433
                    6452  6485  6508  6544  6569  6600  6619  6649  6668  6698  6717  6747  6766
                    6796  6815  6845  6864  6894  6913  6943  6962  7000  7018  7049  7069  7124
                    7149  7193  7214  7377  7378  7411  7412  7444  7445  7637  7654  7668  7705
                    7746  7752  7792  7798  7838  7844  7884  7890  7930  7936  7976  7982  8022
                    8028  8068  8074  8114  8120  8162  8168  8210  8216  8258  8264  8305  8311
                    8352  8358  8399  8405  8447  8453  8494  8500  8543  8549  8590  8596  8637
                    8643  8684  8690  8731  8737  8778  8784  8825  8831  8872  8878  8919  8925
                    8967  8973  9014  9020  9061  9067  9108  9114  9155  9161  9202  9208  9296#
.MSTCL  003142      5206# 5894  5932  5969  5998  6026  6053  6091  6131  6172  6211  6247  6270
                    6320  6369  6418  6467  6524  6585  6634  6683  6732  6781  6830  6879  6928
                    6977  7035  7103  7169  7255  7281  7307  7318  7345  7390  7424  7457  7512
                    7553  7597  7629  7685  7722  7768  7814  7860  7906  7952  7998  8044  8090
                    8138  8186  8234  8281  8328  8375  8423  8470  8519  8566  8613  8660  8707
                    8754  8801  8848  8895  8943  8990  9037  9084  9131  9178
.ROMCL  003230      5221# 5230  5232  5234  5240  5247  5254  5261  5269  5283  5344  5362  5364
                    5367  5372  5376  5403  5414  5416  5424  5426  6278  6280  6297  6299  6327
                    6329  6346  6348  6376  6378  6395  6397  6425  6427  6444  6446  6475  6477
                    6498  6500  6532  6534  6557  6559  6592  6594  6611  6613  6641  6643  6660
                    6662  6690  6692  6709  6711  6739  6741  6758  6760  6788  6790  6807  6809
                    6837  6839  6856  6858  6886  6888  6905  6907  6935  6937  6954  6956  6986
                    6994  7012  7041  7043  7060  7062  7079  7111  7115  7117  7136  7140  7142
                    7178  7183  7186  7205  7207  7237  7262  7296  7334  7366  7369  7399  7403
                    7405  7433  7436  7467  7489  7491  7528  7530  7561  7564  7567  7578  7604
                    7607  7610  7646  7692  7694  7696  7698  7733  7737  7739  7779  7783  7785
                    7825  7829  7831  7871  7875  7877  7917  7921  7923  7963  7967  7969  8009
                    8013  8015  8055  8059  8061  8101  8105  8107  8149  8153  8155  8197  8201
                    8203  8245  8249  8251  8292  8296  8298  8339  8343  8345  8386  8390  8392
                    8434  8438  8440  8481  8485  8487  8530  8534  8536  8577  8581  8583  8624
                    8628  8630  8671  8675  8677  8718  8722  8724  8765  8769  8771  8812  8816
                    8818  8859  8863  8865  8906  8910  8912  8954  8958  8960  9001  9005  9007
                    9048  9052  9054  9095  9099  9101  9142  9146  9148  9189  9193  9195
```

```
BADHEA   5174#  5833  5836  5859  5861  5873  5889  5925  5929  5962  5966  5991  5995  6019  6023
          6047  6050  6082  6086  6123  6127  6164  6168  6203  6207  6237  6243  6263  6267  6313
          6317  6362  6366  6411  6415  6460  6464  6517  6521  6578  6582  6627  6631  6676  6680
          6725  6729  6774  6778  6823  6827  6872  6876  6921  6925  6970  6974  7027  7031  7095
          7099  7162  7166  7222  7225  7248  7251  7273  7277  7310  7314  7348  7351  7383  7386
          7417  7420  7450  7454  7500  7508  7544  7549  7589  7593  7621  7625  7678  7681  7712
          7718  7758  7764  7804  7810  7850  7856  7896  7902  7942  7948  7988  7994  8034  8040
          8080  8086  8128  8134  8175  8181  8224  8230  8271  8277  8318  8324  8365  8371  8412
          8419  8460  8466  8507  8513  8556  8562  8603  8609  8650  8656  8697  8704  8744  8750
          8791  8797  8838  8844  8885  8891  8932  8938  8980  8986  9027  9033  9074  9080  9120
          9126  9168  9174
BCOMPL     102#  4807#  5653  5656  5659
BERROR     110#  4807#  5162#  6285  6304  6334  6353  6383  6402  6432  6451  6484  6507  6542  6568
          6599  6618  6648  6667  6697  6716  6746  6765  6795  6814  6844  6863  6893  6912  6942
          6961  6999  7017  7048  7068  7084
BGNAU      118#  4807#  5819
BGNAUT     132#  4807#  5756
BGNCLN     146#  4807#  5784
BGNDU      159#  4807#  5801
BGNHRD     172#  4807#  9236
BGNHW      187#  4807#  4894
BGNINI     203#  4807#  5638
BGNMOD     215#  4807#  4813
BGNMSG     232#  4807#  5556  5557  5558  5559  5560  5561  5562  5563  5564  5565  5566  5568  5570
          5571  5572  5573  5574  5575  5576  5577  5578  5579  5580  5581  5582  5583  5584  5585
          5586  5587  5588  5589  5591
BGNPRO     245#  4807#  4857
BGNPTA     259#  4807#
BGNRPT     297#  4807#  5607
BGNSEG     311#  4807#  5896  5911  5935  5970  5980  5999  6009  6027  6037  6056  6094  6107  6134
          6147  6174  6189  6213  6224  6275  6293  6324  6342  6373  6391  6422  6440  6471  6493
          6528  6552  6589  6607  6638  6656  6687  6705  6736  6754  6785  6803  6834  6852  6883
          6901  6932  6950  6981  7008  7038  7056  7107  7130  7173  7201  7690  7729  7775  7821
          7867  7913  7959  8005  8051  8097  8145  8193  8241  8288  8335  8382  8430  8477  8526
          8573  8620  8667  8714  8761  8808  8855  8902  8950  8997  9044  9091  9138  9185
BGNSET     325#  4807#
BGNSFT     352#  4807#  9279
BGNSRV     367#  4807#
BGNSUB     380#  4807#  7106
BGNSW      408#  4807#  4923
BGNTST     423#  4807#  5838  5863  5891  5931  5968  5997  6025  6052  6089  6129  6170  6209  6245
          6269  6319  6368  6417  6466  6523  6584  6633  6682  6731  6780  6829  6878  6927  6976
          7033  7101  7168  7227  7253  7279  7316  7353  7388  7422  7456  7510  7551  7595  7627
          7683  7720  7766  7812  7858  7904  7950  7996  8042  8088  8136  8183  8232  8279  8326
          8373  8421  8468  8517  8564  8611  8658  8705  8752  8799  8846  8893  8941  8988  9035
          9082  9128  9176
BNCOMP     457#  4807#  5663  5676
BNERRO     465#  4807#
BREAK      473#  4807#
BRESET     481#  4807#  5785  5803  7229  7304  7354
CKLOOP     491#  4807#  7085
CLOCK      503#  4807#
CLOSE      514#  4807#
CLRVEC     528#  4807#
COMMEN     537#  4807#
DELAY      568#  4807#
```

```
DESCRI     558#  4807#  4983
DEVTYP     588#  4807#  5116
DISPAT     597#  4807#  4871
DISPLA     615#  4807#
DOCLN      635#  4807#
DODU       643#  4807#  5770
DORPT      651#  4807#
ED$CAL    5169#  5833   5836  5859  5861  5873  5889  5925  5929  5962  5966  5991  5995  6019  6023
          6047   6050   6082  6086  6123  6127  6164  6168  6203  6207  6237  6243  6263  6267  6313
          6317   6362   6366  6411  6415  6460  6464  6517  6521  6578  6582  6627  6631  6676  6680
          6725   6729   6774  6778  6823  6827  6872  6876  6921  6925  6970  6974  7027  7031  7095
          7099   7162   7166  7222  7225  7248  7251  7273  7277  7310  7314  7348  7351  7383  7386
          7417   7420   7450  7454  7500  7508  7544  7549  7589  7593  7621  7625  7678  7681  7712
          7718   7758   7764  7804  7810  7850  7856  7896  7902  7942  7948  7988  7994  8034  8040
          8080   8086   8128  8134  8175  8181  8224  8230  8271  8277  8318  8324  8365  8371  8412
          8419   8460   8466  8507  8513  8556  8562  8603  8609  8650  8656  8697  8704  8744  8750
          8791   8797   8838  8844  8885  8891  8932  8938  8980  8986  9027  9033  9074  9080  9120
          9126   9168   9174
ENDAU      659#  4807#  5820
ENDAUT     675#  4807#  5774
ENDCLN     692#  4807#  5787
ENDCOM     708#  4807#
ENDDU      730#  4807#  5804
ENDHRD     749#  4807#  9247
ENDHW      765#  4807#  4909
ENDINI     779#  4807#  5753
ENDMOD     796#  4807#  9290
ENDMSG     813#  4807#  5556  5557  5558  5559  5560  5561  5562  5563  5564  5565  5566  5568  5570
          5571   5572   5573  5574  5575  5576  5577  5578  5579  5580  5581  5582  5583  5584  5585
          5586   5587   5588  5589  5594
ENDPRO     830#  4807#  4861
ENDPTA     844#  4807#
ENDRPT     857#  4807#  5624
ENDSEG     876#  4807#  5908  5922  5952  5979  5987  6008  6015  6036  6043  6074  6105  6120  6145
          6161   6188   6200  6222  6231  6290  6310  6339  6359  6388  6408  6437  6457  6490  6514
          6549   6575   6604  6624  6653  6673  6702  6722  6751  6771  6800  6820  6849  6869  6898
          6918   6947   6967  7005  7024  7053  7092  7128  7154  7198  7219  7706  7751  7797  7843
          7889   7935   7981  8027  8073  8119  8167  8215  8263  8310  8357  8404  8452  8499  8548
          8595   8642   8689  8736  8783  8830  8877  8924  8972  9019  9066  9113  9160  9207
ENDSET     895#  4807#
ENDSFT     915#  4807#  9282
ENDSRV     932#  4807#
ENDSUB     952#  4807#  7159
ENDSW      974#  4807#  4926
ENDTST     988#  4807#  5857  5871  5923  5953  5989  6017  6045  6075  6121  6162  6201  6232  6261
          6311   6360   6409  6458  6515  6576  6625  6674  6723  6772  6821  6870  6919  6968  7025
          7093   7160   7220  7246  7271  7308  7346  7381  7415  7448  7498  7542  7587  7619  7672
          7707   7756   7802  7848  7894  7940  7986  8032  8078  8125  8172  8221  8268  8315  8362
          8409   8457   8504  8553  8600  8647  8694  8741  8788  8835  8882  8929  8977  9024  9071
          9118   9165   9212
EQUALS    1009#  4807#  4953
ERRDF     1087#  4807#  5383  5852  5869  5902  5916  5942  5948  5977  5985  6006  6014  6034  6042
          6064   6070   6100  6114  6140  6155  6180  6194  6220  6229  6257  6285  6304  6334  6353
          6383   6402   6432  6451  6484  6507  6542  6568  6599  6618  6648  6667  6697  6716  6746
          6765   6795   6814  6844  6863  6893  6912  6942  6961  6999  7017  7048  7068  7084  7123
          7148   7192   7213  7241  7243  7266  7268  7305  7338  7342  7376  7410  7443  7496  7537
```

```
          7574    7584    7617    7653    7704    7745    7791    7837    7883    7929    7975    8021    8067    8113    8161
          8209    8257    8304    8351    8398    8446    8493    8542    8589    8636    8683    8730    8777    8824    8871
          8918    8966    9013    9060    9107    9154    9201
ERRHRD    1099#   4807#
ERROR     1109#   4807#   5147#   5383    5852    5869    5902    5916    5942    5948    5977    5985    6006    6014    6034
          6042    6064    6070    6100    6114    6140    6155    6180    6194    6220    6229    6257    7241    7243    7266
          7268    7305    7338    7342    7376    7410    7443    7496    7537    7574    7584    7617    7653    7704    7745
          7791    7837    7883    7929    7975    8021    8067    8113    8161    8209    8257    8304    8351    8398    8446
          8493    8542    8589    8636    8683    8730    8777    8824    8871    8918    8966    9013    9060    9107    9154
          9201
ERRSF     1118#   4807#
ERRSOF    1130#   4807#
ERRTBL    1140#   4807#
ESCAPE    1156#   4807#   5845    5856    5903    5917    5943    5949    5978    5986    6007    6035    6065    6071    6101
          6115    6141    6156    6181    6195    6221    6230    6259    6286    6305    6335    6354    6384    6403    6433
          6452    6485    6508    6544    6569    6600    6619    6649    6668    6698    6717    6747    6766    6796    6815
          6845    6864    6894    6913    6943    6962    7000    7018    7049    7069    7124    7149    7193    7214    7377
          7411    7444    7654    7705    7746    7792    7838    7884    7930    7976    8022    8068    8114    8162    8210
          8258    8305    8352    8399    8447    8494    8543    8590    8637    8684    8731    8778    8825    8872    8919
          8967    9014    9061    9108    9155    9202
EXIT      1186#   4807#   5615    7378    7412    7445    7668    7752    7798    7844    7890    7936    7982    8028    8074
          8120    8168    8216    8264    8311    8358    8405    8453    8500    8549    8596    8643    8690    8737    8784
          8831    8878    8925    8973    9020    9067    9114    9161    9208
FEQUAL    1228#   4807#
GETBYT    1246#   4807#
GETPRI    1264#   4807#
GETWOR    1256#   4807#
GMANIA    1286#   4807#
GMANID    1299#   4807#
GMANIL    1315#   4807#
GPHARD    1328#   4807#   5675
GPRMA     1340#   4807#   9239    9240
GPRMD     1372#   4807#   9238    9241
GPRML     1407#   4807#
HEADER    1432#   4807#   4848
INLOOP    1446#   4807#
IOSETU    1453#   4807#
IOSTAR    1466#   4807#
KT11      1488#   4807#
K4ONLY    5179#
LASTAD    1659#   4807#   9298
MANUAL    1677#   4807#
MDT0      5520#   5566    5571    5572    5574    5575    5577    5584    5585
MDT1      5523#   5559    5560    5561    5562    5563
MDT2      5527#   5556    5557    5558    5564    5565    5568    5573    5578    5579    5581    5583    5586    5587    5588
          5589
MDT27     5536#   5580    5582
MDT5      5532#   5570    5576
MEMORY    1685#   4807#
MSTCLR    5200#   5894    5932    5969    5998    6026    6053    6091    6131    6172    6211    6247    6270    6320    6369
          6418    6467    6524    6585    6634    6683    6732    6781    6830    6879    6928    6977    7035    7103    7169
          7255    7281    7307    7318    7345    7390    7424    7457    7512    7553    7597    7629    7685    7722    7768
          7814    7860    7906    7952    7998    8044    8090    8138    8186    8234    8281    8328    8375    8423    8470
          8519    8566    8613    8660    8707    8754    8801    8848    8895    8943    8990    9037    9084    9131    9178
MYINT     5188#   5864    5892    6171    6212    6246    6274    6323    6372    6421    6470    6527    6588    6637    6686
          6735    6784    6833    6882    6931    6980    7037    7102    7172    7228    7254    7280    7317    7356    7389
```

	7423	7458	7511	7552	7596	7628	7684	7721	7767	7813	7859	7905	7951	7997	8043
	8089	8137	8184	8233	8280	8327	8374	8422	8469	8518	8565	8612	8659	8706	8753
	8800	8847	8894	8942	8989	9036	9083	9130	9177						
M$BYTE	2901#	4807#	4848#												
M$CHEC	3206#	4807#	5615#	7378#	7412#	7445#	7668#	7752#	7798#	7844#	7890#	7936#	7982#	8028#	8074#
	8120#	8168#	8216#	8264#	8311#	8358#	8405#	8453#	8500#	8549#	8596#	8643#	8690#	8737#	8784#
	8831#	8878#	8925#	8973#	9020#	9067#	9114#	9161#	9208#						
M$CNTO	3279#	4807#	9238#	9239#	9240#	9241#									
M$COUN	3124#	4807#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#	5568#	5570#
	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#	5584#	5585#
	5586#	5587#	5588#	5589#	5592#	5593#									
M$DATA	2614#	4807#	4848#	4983#	5116#										
M$DECR	3063#	4807#	4861#	4909#	4926#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#
	5566#	5568#	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#
	5583#	5584#	5585#	5586#	5587#	5588#	5589#	5594#	5624#	5753#	5774#	5787#	5804#	5820#	5857#
	5871#	5908#	5922#	5923#	5952#	5953#	5979#	5987#	5989#	6008#	6015#	6017#	6036#	6043#	6045#
	6074#	6075#	6105#	6120#	6121#	6145#	6161#	6162#	6188#	6200#	6201#	6222#	6231#	6232#	6261#
	6290#	6310#	6311#	6339#	6359#	6360#	6388#	6408#	6409#	6437#	6457#	6458#	6490#	6514#	6515#
	6549#	6575#	6576#	6604#	6624#	6625#	6653#	6673#	6674#	6702#	6722#	6723#	6751#	6771#	6772#
	6800#	6820#	6821#	6849#	6869#	6870#	6898#	6918#	6919#	6947#	6967#	6968#	7005#	7024#	7025#
	7053#	7092#	7093#	7128#	7154#	7159#	7160#	7198#	7219#	7220#	7246#	7271#	7308#	7346#	7381#
	7415#	7448#	7498#	7542#	7587#	7619#	7672#	7706#	7707#	7751#	7756#	7797#	7802#	7843#	7848#
	7889#	7894#	7935#	7940#	7981#	7986#	8027#	8032#	8073#	8078#	8119#	8125#	8167#	8172#	8215#
	8221#	8263#	8268#	8310#	8315#	8357#	8362#	8404#	8409#	8452#	8457#	8499#	8504#	8548#	8553#
	8595#	8600#	8642#	8647#	8689#	8694#	8736#	8741#	8783#	8788#	8830#	8835#	8877#	8882#	8924#
	8929#	8972#	8977#	9019#	9024#	9066#	9071#	9113#	9118#	9160#	9165#	9207#	9212#	9247#	9282#
	9290#														
M$DEFA	3263#	4807#	9238#	9239#	9240#	9241#									
M$ENDE	3145#	4807#	4909#	4926#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#
	5568#	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#
	5584#	5585#	5586#	5587#	5588#	5589#	5594#	5624#	5753#	5774#	5787#	5804#	5820#	5857#	5871#
	5908#	5922#	5923#	5952#	5953#	5979#	5987#	5989#	6008#	6015#	6017#	6036#	6043#	6045#	6074#
	6075#	6105#	6120#	6121#	6145#	6161#	6162#	6188#	6200#	6201#	6222#	6231#	6232#	6261#	6290#
	6310#	6311#	6339#	6359#	6360#	6388#	6408#	6409#	6437#	6457#	6458#	6490#	6514#	6515#	6549#
	6575#	6576#	6604#	6624#	6625#	6653#	6673#	6674#	6702#	6722#	6723#	6751#	6771#	6772#	6800#
	6820#	6821#	6849#	6869#	6870#	6898#	6918#	6919#	6947#	6967#	6968#	7005#	7024#	7025#	7053#
	7092#	7093#	7128#	7154#	7159#	7160#	7198#	7219#	7220#	7246#	7271#	7308#	7346#	7381#	7415#
	7448#	7498#	7542#	7587#	7619#	7672#	7706#	7707#	7751#	7756#	7797#	7802#	7843#	7848#	7889#
	7894#	7935#	7940#	7981#	7986#	8027#	8032#	8073#	8078#	8119#	8125#	8167#	8172#	8215#	8221#
	8263#	8268#	8310#	8315#	8357#	8362#	8404#	8409#	8452#	8457#	8499#	8504#	8548#	8553#	8595#
	8600#	8642#	8647#	8689#	8694#	8736#	8741#	8783#	8788#	8830#	8835#	8877#	8882#	8924#	8929#
	8972#	8977#	9019#	9024#	9066#	9071#	9113#	9118#	9160#	9165#	9207#	9212#	9247#	9282#	9290#
M$ERRI	2365#	4807#	5383#	5852#	5869#	5902#	5916#	5942#	5948#	5977#	5985#	6006#	6014#	6034#	6042#
	6064#	6070#	6100#	6114#	6140#	6155#	6180#	6194#	6220#	6229#	6257#	6285#	6304#	6334#	6353#
	6383#	6402#	6432#	6451#	6484#	6507#	6542#	6568#	6599#	6618#	6648#	6667#	6697#	6716#	6746#
	6765#	6795#	6814#	6844#	6863#	6893#	6912#	6942#	6961#	6999#	7017#	7048#	7068#	7084#	7123#
	7148#	7192#	7213#	7241#	7243#	7266#	7268#	7305#	7338#	7342#	7376#	7410#	7443#	7496#	7537#
	7574#	7584#	7617#	7653#	7704#	7745#	7791#	7837#	7883#	7929#	7975#	8021#	8067#	8113#	8161#
	8209#	8257#	8304#	8351#	8398#	8446#	8493#	8542#	8589#	8636#	8683#	8730#	8777#	8824#	8871#
	8918#	8966#	9013#	9060#	9107#	9154#	9201#								
M$ESCA	2921#	4807#	5845#	5856#	5903#	5917#	5943#	5949#	5978#	5986#	6007#	6035#	6065#	6071#	6101#
	6115#	6141#	6156#	6181#	6195#	6221#	6230#	6259#	6286#	6305#	6335#	6354#	6384#	6403#	6433#
	6452#	6485#	6508#	6544#	6569#	6600#	6619#	6649#	6668#	6698#	6717#	6747#	6766#	6796#	6815#
	6845#	6864#	6894#	6913#	6943#	6962#	7000#	7018#	7049#	7069#	7124#	7149#	7193#	7214#	7377#
	7411#	7444#	7654#	7705#	7746#	7792#	7838#	7884#	7930#	7976#	8022#	8068#	8114#	8162#	8210#
	8258#	8305#	8352#	8399#	8447#	8494#	8543#	8590#	8637#	8684#	8731#	8778#	8825#	8872#	8919#

```
                8967#   9014#   9061#   9108#   9155#   9202#
M$ESCS          2932#   4807#   5845#   5856#   5903#   5917#   5943#   5949#   5978#   5986#   6007#   6035#   6065#   6071#   6101#
                6115#   6141#   6156#   6181#   6195#   6221#   6230#   6259#   6286#   6305#   6335#   6354#   6384#   6403#   6433#
                6452#   6485#   6508#   6544#   6569#   6600#   6619#   6649#   6668#   6698#   6717#   6747#   6766#   6796#   6815#
                6845#   6864#   6894#   6913#   6943#   6962#   7000#   7018#   7049#   7069#   7124#   7149#   7193#   7214#   7377#
                7411#   7444#   7654#   7705#   7746#   7792#   7838#   7884#   7930#   7976#   8022#   8068#   8114#   8162#   8210#
                8258#   8305#   8352#   8399#   8447#   8494#   8543#   8590#   8637#   8684#   8731#   8778#   8825#   8872#   8919#
                8967#   9014#   9061#   9108#   9155#   9202#
M$EXCP          3186#   4807#   9238#   9239#   9240#   9241#
M$EXIT          2943#   4807#   5615#   7378#   7412#   7445#   7668#   7752#   7798#   7844#   7890#   7936#   7982#   8028#   8074#
                8120#   8168#   8216#   8264#   8311#   8358#   8405#   8453#   8500#   8549#   8596#   8643#   8690#   8737#   8784#
                8831#   8878#   8925#   8973#   9020#   9067#   9114#   9161#   9208#
M$EXSE          2965#   4807#   5615#   7378#   7412#   7445#   7668#   7752#   7798#   7844#   7890#   7936#   7982#   8028#   8074#
                8120#   8168#   8216#   8264#   8311#   8358#   8405#   8453#   8500#   8549#   8596#   8643#   8690#   8737#   8784#
                8831#   8878#   8925#   8973#   9020#   9067#   9114#   9161#   9208#
M$EXTJ          2954#   4807#   5615#   7378#   7412#   7445#   7668#   7752#   7798#   7844#   7890#   7936#   7982#   8028#   8074#
                8120#   8168#   8216#   8264#   8311#   8358#   8405#   8453#   8500#   8549#   8596#   8643#   8690#   8737#   8784#
                8831#   8878#   8925#   8973#   9020#   9067#   9114#   9161#   9208#
M$GEN           3087#   4807#   4813#   4848#   4857#   4871#   4894#   4909#   4923#   4926#   4983#   5116#   5556#   5557#   5558#
                5559#   5560#   5561#   5562#   5563#   5564#   5565#   5566#   5568#   5570#   5571#   5572#   5573#   5574#   5575#
                5576#   5577#   5578#   5579#   5580#   5581#   5582#   5583#   5584#   5585#   5586#   5587#   5588#   5589#   5591#
                5594#   5607#   5624#   5638#   5753#   5756#   5774#   5784#   5787#   5801#   5804#   5819#   5820#   5838#   5857#
                5863#   5871#   5891#   5908#   5922#   5923#   5931#   5952#   5953#   5968#   5979#   5987#   5989#   5997#   6008#
                6015#   6017#   6025#   6036#   6043#   6045#   6052#   6074#   6075#   6089#   6105#   6120#   6121#   6129#   6145#
                6161#   6162#   6170#   6188#   6200#   6201#   6209#   6222#   6231#   6232#   6245#   6261#   6269#   6290#   6310#
                6311#   6319#   6339#   6359#   6360#   6368#   6388#   6408#   6409#   6417#   6437#   6457#   6458#   6466#   6490#
                6514#   6515#   6523#   6549#   6575#   6576#   6584#   6604#   6624#   6625#   6633#   6653#   6673#   6674#   6682#
                6702#   6722#   6723#   6731#   6751#   6771#   6772#   6780#   6800#   6820#   6821#   6829#   6849#   6869#   6870#
                6878#   6898#   6918#   6919#   6927#   6947#   6967#   6968#   6976#   7005#   7024#   7025#   7033#   7053#   7092#
                7093#   7101#   7106#   7128#   7154#   7159#   7160#   7168#   7198#   7219#   7220#   7227#   7246#   7253#   7271#
                7279#   7308#   7316#   7346#   7353#   7381#   7388#   7415#   7422#   7448#   7456#   7498#   7510#   7542#   7551#
                7587#   7595#   7619#   7627#   7672#   7683#   7706#   7707#   7720#   7751#   7756#   7766#   7797#   7802#   7812#
                7843#   7848#   7858#   7889#   7894#   7904#   7935#   7940#   7950#   7981#   7986#   7996#   8027#   8032#   8042#
                8073#   8078#   8088#   8119#   8125#   8136#   8167#   8172#   8183#   8215#   8221#   8232#   8263#   8268#   8279#
                8310#   8315#   8326#   8357#   8362#   8373#   8404#   8409#   8421#   8452#   8457#   8468#   8499#   8504#   8517#
                8548#   8553#   8564#   8595#   8600#   8611#   8642#   8647#   8658#   8689#   8694#   8705#   8736#   8741#   8752#
                8783#   8788#   8799#   8830#   8835#   8846#   8877#   8882#   8893#   8924#   8929#   8941#   8972#   8977#   8988#
                9019#   9024#   9035#   9066#   9071#   9082#   9113#   9118#   9128#   9160#   9165#   9176#   9207#   9212#   9236#
                9247#   9279#   9282#   9298#
M$GENB          2764#   4807#
M$GETS          3079#   4807#   4861#   4909#   4926#   5556#   5557#   5558#   5559#   5560#   5561#   5562#   5563#   5564#   5565#
                5566#   5568#   5570#   5571#   5572#   5573#   5574#   5575#   5576#   5577#   5578#   5579#   5580#   5581#   5582#
                5583#   5584#   5585#   5586#   5587#   5588#   5589#   5594#   5624#   5753#   5774#   5787#   5804#   5820#   5857#
                5871#   5903#   5908#   5917#   5922#   5923#   5943#   5949#   5952#   5953#   5978#   5979#   5986#   5987#   5989#
                6007#   6008#   6015#   6017#   6035#   6036#   6043#   6045#   6065#   6071#   6074#   6075#   6101#   6105#   6115#
                6120#   6121#   6141#   6145#   6156#   6161#   6162#   6181#   6188#   6195#   6200#   6201#   6221#   6222#   6230#
                6231#   6232#   6261#   6286#   6290#   6305#   6310#   6311#   6335#   6339#   6354#   6359#   6360#   6384#   6388#
                6403#   6408#   6409#   6433#   6437#   6452#   6457#   6458#   6485#   6490#   6508#   6514#   6515#   6544#   6549#
                6569#   6575#   6576#   6600#   6604#   6619#   6624#   6625#   6649#   6653#   6668#   6673#   6674#   6698#   6702#
                6717#   6722#   6723#   6747#   6751#   6766#   6771#   6772#   6796#   6800#   6815#   6820#   6821#   6845#   6849#
                6864#   6869#   6870#   6894#   6898#   6913#   6918#   6919#   6943#   6947#   6962#   6967#   6968#   7000#   7005#
                7018#   7024#   7025#   7049#   7053#   7069#   7092#   7093#   7124#   7128#   7154#   7159#   7160#   7193#   7198#
                7214#   7219#   7220#   7246#   7271#   7308#   7346#   7381#   7415#   7448#   7498#   7542#   7587#   7619#   7672#
                7705#   7706#   7707#   7746#   7751#   7756#   7792#   7797#   7802#   7838#   7843#   7848#   7884#   7889#   7894#
                7930#   7935#   7940#   7976#   7981#   7986#   8022#   8027#   8032#   8068#   8073#   8078#   8114#   8119#   8125#
                8162#   8167#   8172#   8210#   8215#   8221#   8258#   8263#   8268#   8305#   8310#   8315#   8352#   8357#   8362#
```

```
                8399#  8404#  8409#  8447#  8452#  8457#  8494#  8499#  8504#  8543#  8548#  8553#  8590#  8595#  8600#
                8637#  8642#  8647#  8684#  8689#  8694#  8731#  8736#  8741#  8778#  8783#  8788#  8825#  8830#  8835#
                8872#  8877#  8882#  8919#  8924#  8929#  8967#  8972#  8977#  9014#  9019#  9024#  9061#  9066#  9071#
                9108#  9113#  9118#  9155#  9160#  9165#  9202#  9207#  9212#  9247#  9282#  9290#
M$GETT          2634#  4807#  5615#  5845#  5856#  5903#  5917#  5943#  5949#  5978#  5986#  6007#  6035#  6065#  6071#
                6101#  6115#  6141#  6156#  6181#  6195#  6221#  6230#  6259#  6286#  6305#  6335#  6354#  6384#  6403#
                6433#  6452#  6485#  6508#  6544#  6569#  6600#  6619#  6649#  6668#  6698#  6717#  6747#  6766#  6796#
                6815#  6845#  6864#  6894#  6913#  6943#  6962#  7000#  7018#  7049#  7069#  7124#  7149#  7193#  7214#
                7377#  7378#  7411#  7412#  7444#  7445#  7654#  7668#  7705#  7746#  7752#  7792#  7798#  7838#  7844#
                7884#  7890#  7930#  7936#  7976#  7982#  8022#  8028#  8068#  8074#  8114#  8120#  8162#  8168#  8210#
                8216#  8258#  8264#  8305#  8311#  8352#  8358#  8399#  8405#  8447#  8453#  8494#  8500#  8543#  8549#
                8590#  8596#  8637#  8643#  8684#  8690#  8731#  8737#  8778#  8784#  8825#  8831#  8872#  8878#  8919#
                8925#  8967#  8973#  9014#  9020#  9061#  9067#  9108#  9114#  9155#  9161#  9202#  9208#
M$GNGB          2689#  4807#  4813#  4848#  4857#  4871#  4894#  4923#  4983#  5116#  5556#  5557#  5558#  5559#  5560#
                5561#  5562#  5563#  5564#  5565#  5566#  5568#  5570#  5571#  5572#  5573#  5574#  5575#  5576#  5577#
                5578#  5579#  5580#  5581#  5582#  5583#  5584#  5585#  5586#  5587#  5588#  5589#  5591#  5607#  5638#
                5756#  5784#  5801#  5819#  9236#  9279#  9298#
M$GNIN          3101#  4807#  4848#  4871#  4894#  4923#  4983#  5116#  5383#  5556#  5557#  5558#  5559#  5560#  5561#
                5562#  5563#  5564#  5565#  5566#  5568#  5570#  5571#  5572#  5573#  5574#  5575#  5576#  5577#  5578#
                5579#  5580#  5581#  5582#  5583#  5584#  5585#  5586#  5587#  5588#  5589#  5592#  5593#  5594#  5615#
                5624#  5652#  5653#  5655#  5656#  5658#  5659#  5662#  5663#  5675#  5676#  5753#  5770#  5774#  5785#
                5787#  5803#  5804#  5820#  5845#  5852#  5856#  5857#  5869#  5871#  5896#  5902#  5903#  5908#  5911#
                5916#  5917#  5922#  5923#  5935#  5942#  5943#  5948#  5949#  5952#  5953#  5970#  5977#  5978#  5979#
                5980#  5985#  5986#  5987#  5989#  5999#  6006#  6007#  6008#  6009#  6014#  6015#  6017#  6027#  6034#
                6035#  6036#  6037#  6042#  6043#  6045#  6056#  6064#  6065#  6070#  6071#  6074#  6075#  6094#  6100#
                6101#  6105#  6107#  6114#  6115#  6120#  6121#  6134#  6140#  6141#  6145#  6147#  6155#  6156#  6161#
                6162#  6174#  6180#  6181#  6188#  6189#  6194#  6195#  6200#  6201#  6213#  6220#  6221#  6222#  6224#
                6229#  6230#  6231#  6232#  6257#  6259#  6261#  6275#  6285#  6286#  6290#  6293#  6304#  6305#  6310#
                6311#  6324#  6334#  6335#  6339#  6342#  6353#  6354#  6359#  6360#  6373#  6383#  6384#  6388#  6391#
                6402#  6403#  6408#  6409#  6422#  6432#  6433#  6437#  6440#  6451#  6452#  6457#  6458#  6471#  6484#
                6485#  6490#  6493#  6507#  6508#  6514#  6515#  6528#  6542#  6544#  6549#  6552#  6568#  6569#  6575#
                6576#  6589#  6599#  6600#  6604#  6607#  6618#  6619#  6624#  6625#  6638#  6648#  6649#  6653#  6656#
                6667#  6668#  6673#  6674#  6687#  6697#  6698#  6702#  6705#  6716#  6717#  6722#  6723#  6736#  6746#
                6747#  6751#  6754#  6765#  6766#  6771#  6772#  6785#  6795#  6796#  6800#  6803#  6814#  6815#  6820#
                6821#  6834#  6844#  6845#  6849#  6852#  6863#  6864#  6869#  6870#  6883#  6893#  6894#  6898#  6901#
                6912#  6913#  6918#  6919#  6932#  6942#  6943#  6947#  6950#  6961#  6962#  6967#  6968#  6981#  6999#
                7000#  7005#  7008#  7017#  7018#  7024#  7025#  7038#  7048#  7049#  7053#  7056#  7068#  7069#  7084#
                7085#  7092#  7093#  7106#  7107#  7123#  7124#  7128#  7130#  7148#  7149#  7154#  7159#  7160#  7173#
                7192#  7193#  7198#  7201#  7213#  7214#  7219#  7220#  7229#  7235#  7239#  7241#  7243#  7246#  7260#
                7264#  7266#  7268#  7271#  7283#  7298#  7304#  7305#  7308#  7320#  7336#  7338#  7342#  7346#  7354#
                7376#  7377#  7378#  7381#  7410#  7411#  7412#  7415#  7443#  7444#  7445#  7448#  7496#  7498#  7537#
                7542#  7574#  7584#  7587#  7617#  7619#  7653#  7654#  7668#  7672#  7690#  7704#  7705#  7706#  7707#
                7729#  7745#  7746#  7751#  7752#  7756#  7775#  7791#  7792#  7797#  7798#  7802#  7821#  7837#  7838#
                7843#  7844#  7848#  7867#  7883#  7884#  7889#  7890#  7894#  7913#  7929#  7930#  7935#  7936#  7940#
                7959#  7975#  7976#  7981#  7982#  7986#  8005#  8021#  8022#  8027#  8028#  8032#  8051#  8067#  8068#
                8073#  8074#  8078#  8097#  8113#  8114#  8119#  8120#  8125#  8145#  8161#  8162#  8167#  8168#  8172#
                8193#  8209#  8210#  8215#  8216#  8221#  8241#  8257#  8258#  8263#  8264#  8268#  8288#  8304#  8305#
                8310#  8311#  8315#  8335#  8351#  8352#  8357#  8358#  8362#  8382#  8398#  8399#  8404#  8405#  8409#
                8430#  8446#  8447#  8452#  8453#  8457#  8477#  8493#  8494#  8499#  8500#  8504#  8526#  8542#  8543#
                8548#  8549#  8553#  8573#  8589#  8590#  8595#  8596#  8600#  8620#  8636#  8637#  8642#  8643#  8647#
                8667#  8683#  8684#  8689#  8690#  8694#  8714#  8730#  8731#  8736#  8737#  8741#  8761#  8777#  8778#
                8783#  8784#  8788#  8808#  8824#  8825#  8830#  8831#  8835#  8855#  8871#  8872#  8877#  8878#  8882#
                8902#  8918#  8919#  8924#  8925#  8929#  8950#  8966#  8967#  8972#  8973#  8977#  8997#  9013#  9014#
                9019#  9020#  9024#  9044#  9060#  9061#  9066#  9067#  9071#  9091#  9107#  9108#  9113#  9114#  9118#
                9138#  9154#  9155#  9160#  9161#  9165#  9185#  9201#  9202#  9207#  9208#  9212#  9236#  9238#  9239#
                9240#  9241#  9247#  9279#  9282#  9298#
```

M$GNLS 2717# 4807# 5908# 5922# 5952# 5979# 5987# 6008# 6015# 6036# 6043# 6074# 6105# 6120# 6145#
6161# 6188# 6200# 6222# 6231# 6290# 6310# 6339# 6359# 6388# 6408# 6437# 6457# 6490# 6514#
6549# 6575# 6604# 6624# 6653# 6673# 6702# 6722# 6751# 6771# 6800# 6820# 6849# 6869# 6898#
6918# 6947# 6967# 7005# 7024# 7053# 7092# 7128# 7154# 7198# 7219# 7706# 7751# 7797# 7843#
7889# 7935# 7981# 8027# 8073# 8119# 8167# 8215# 8263# 8310# 8357# 8404# 8452# 8499# 8548#
8595# 8642# 8689# 8736# 8783# 8830# 8877# 8924# 8972# 9019# 9066# 9113# 9160# 9207#

M$GNSU 2679# 4807# 7106#

M$GNTA 2659# 4807# 4909# 4926# 5556# 5557# 5558# 5559# 5560# 5561# 5562# 5563# 5564# 5565# 5566#
5568# 5570# 5571# 5572# 5573# 5574# 5575# 5576# 5577# 5578# 5579# 5580# 5581# 5582# 5583#
5584# 5585# 5586# 5587# 5588# 5589# 5594# 5624# 5753# 5774# 5787# 5804# 5820# 5857# 5871#
5923# 5953# 5989# 6017# 6045# 6075# 6121# 6162# 6201# 6232# 6261# 6311# 6360# 6409# 6458#
6515# 6576# 6625# 6674# 6723# 6772# 6821# 6870# 6919# 6968# 7025# 7093# 7159# 7160# 7220#
7246# 7271# 7308# 7346# 7381# 7415# 7448# 7498# 7542# 7587# 7619# 7672# 7707# 7756# 7802#
7848# 7894# 7940# 7986# 8032# 8078# 8125# 8172# 8221# 8268# 8315# 8362# 8409# 8457# 8504#
8553# 8600# 8647# 8694# 8741# 8788# 8835# 8882# 8929# 8977# 9024# 9071# 9118# 9165# 9212#
9247# 9282#

M$GNTE 2669# 4807# 5838# 5863# 5891# 5931# 5968# 5997# 6025# 6052# 6089# 6129# 6170# 6209# 6245#
6269# 6319# 6368# 6417# 6466# 6523# 6584# 6633# 6682# 6731# 6780# 6829# 6878# 6927# 6976#
7033# 7101# 7168# 7227# 7253# 7279# 7316# 7353# 7388# 7422# 7456# 7510# 7551# 7595# 7627#
7683# 7720# 7766# 7812# 7858# 7904# 7950# 7996# 8042# 8088# 8136# 8183# 8232# 8279# 8326#
8373# 8421# 8468# 8517# 8564# 8611# 8658# 8705# 8752# 8799# 8846# 8893# 8941# 8988# 9035#
9082# 9128# 9176#

M$HAPT 2477# 4807# 4848#

M$HNAP 2565# 4807# 4848#

M$INCR 3054# 4807# 4813# 4857# 4894# 4923# 5383# 5556# 5557# 5558# 5559# 5560# 5561# 5562# 5563#
5564# 5565# 5566# 5568# 5570# 5571# 5572# 5573# 5574# 5575# 5576# 5577# 5578# 5579# 5580#
5581# 5582# 5583# 5584# 5585# 5586# 5587# 5588# 5589# 5591# 5592# 5593# 5594# 5607# 5624#
5638# 5652# 5655# 5658# 5662# 5675# 5753# 5756# 5770# 5774# 5784# 5785# 5787# 5801# 5803#
5804# 5819# 5820# 5838# 5845# 5852# 5856# 5857# 5863# 5869# 5871# 5891# 5896# 5902# 5903#
5908# 5911# 5916# 5917# 5922# 5923# 5931# 5935# 5942# 5943# 5948# 5949# 5952# 5953# 5968#
5970# 5977# 5978# 5979# 5980# 5985# 5986# 5987# 5989# 5997# 5999# 6006# 6007# 6008# 6009#
6014# 6015# 6017# 6025# 6027# 6034# 6035# 6036# 6037# 6042# 6043# 6045# 6052# 6056# 6064#
6065# 6070# 6071# 6074# 6075# 6089# 6094# 6100# 6101# 6105# 6107# 6114# 6115# 6120# 6121#
6129# 6134# 6140# 6141# 6145# 6147# 6155# 6156# 6161# 6162# 6170# 6174# 6180# 6181# 6188#
6189# 6194# 6195# 6200# 6201# 6209# 6213# 6220# 6221# 6222# 6224# 6229# 6230# 6231# 6232#
6245# 6257# 6259# 6261# 6269# 6275# 6285# 6286# 6290# 6293# 6304# 6305# 6310# 6311# 6319#
6324# 6334# 6335# 6339# 6342# 6353# 6354# 6359# 6360# 6368# 6373# 6383# 6384# 6388# 6391#
6402# 6403# 6408# 6409# 6417# 6422# 6432# 6433# 6437# 6440# 6451# 6452# 6457# 6458# 6466#
6471# 6484# 6485# 6490# 6493# 6507# 6508# 6514# 6515# 6523# 6528# 6542# 6544# 6549# 6552#
6568# 6569# 6575# 6576# 6584# 6589# 6599# 6600# 6604# 6607# 6618# 6619# 6624# 6625# 6633#
6638# 6648# 6649# 6653# 6656# 6667# 6668# 6673# 6674# 6682# 6687# 6697# 6698# 6702# 6705#
6716# 6717# 6722# 6723# 6731# 6736# 6746# 6747# 6751# 6754# 6765# 6766# 6771# 6772# 6780#
6785# 6795# 6796# 6800# 6803# 6814# 6815# 6820# 6821# 6829# 6834# 6844# 6845# 6849# 6852#
6863# 6864# 6869# 6870# 6878# 6883# 6893# 6894# 6898# 6901# 6912# 6913# 6918# 6919# 6927#
6932# 6942# 6943# 6947# 6950# 6961# 6962# 6967# 6968# 6976# 6981# 6999# 7000# 7005# 7008#
7017# 7018# 7024# 7025# 7033# 7038# 7048# 7049# 7053# 7056# 7068# 7069# 7084# 7085# 7092#
7093# 7101# 7106# 7107# 7123# 7124# 7128# 7130# 7148# 7149# 7154# 7159# 7160# 7168# 7173#
7192# 7193# 7198# 7201# 7213# 7214# 7219# 7220# 7227# 7229# 7235# 7239# 7241# 7243# 7246#
7253# 7260# 7264# 7266# 7268# 7271# 7279# 7283# 7298# 7304# 7305# 7308# 7316# 7320# 7336#
7338# 7342# 7346# 7353# 7354# 7376# 7377# 7378# 7381# 7388# 7410# 7411# 7412# 7415# 7422#
7443# 7444# 7445# 7448# 7456# 7496# 7498# 7510# 7537# 7542# 7551# 7574# 7584# 7587# 7595#
7617# 7619# 7627# 7653# 7654# 7668# 7672# 7683# 7690# 7704# 7705# 7706# 7707# 7720# 7729#
7745# 7746# 7751# 7752# 7756# 7766# 7775# 7791# 7792# 7797# 7798# 7802# 7812# 7821# 7837#
7838# 7843# 7844# 7848# 7858# 7867# 7883# 7884# 7889# 7890# 7894# 7904# 7913# 7929# 7930#
7935# 7936# 7940# 7950# 7959# 7975# 7976# 7981# 7982# 7986# 7996# 8005# 8021# 8022# 8027#
8028# 8032# 8042# 8051# 8067# 8068# 8073# 8074# 8078# 8088# 8097# 8113# 8114# 8119# 8120#

	8125#	8136#	8145#	8161#	8162#	8167#	8168#	8172#	8183#	8193#	8209#	8210#	8215#	8216#	8221#
	8232#	8241#	8257#	8258#	8263#	8264#	8268#	8279#	8288#	6304#	8305#	8310#	8311#	8315#	8326#
	8335#	8351#	8352#	8357#	8358#	8362#	8373#	8382#	8398#	8399#	8404#	8405#	8409#	8421#	8430#
	8446#	8447#	8452#	8453#	8457#	8468#	8477#	8493#	8494#	8499#	8500#	8504#	8517#	8526#	8542#
	8543#	8548#	8549#	8553#	8564#	8573#	8589#	8590#	8595#	8596#	8600#	8611#	8620#	8636#	8637#
	8642#	8643#	8647#	8658#	8667#	8683#	8684#	8689#	8690#	8694#	8705#	8714#	8730#	8731#	8736#
	8737#	8741#	8752#	8761#	8777#	8778#	8783#	8784#	8788#	8799#	8808#	8824#	8825#	8830#	8831#
	8835#	8846#	8855#	8871#	8872#	8877#	8878#	8882#	8893#	8902#	8918#	8919#	8924#	8925#	8929#
	8941#	8950#	8966#	8967#	8972#	8973#	8977#	8988#	8997#	9013#	9014#	9019#	9020#	9024#	9035#
	9044#	9060#	9061#	9066#	9067#	9071#	9082#	9091#	9107#	9108#	9113#	9114#	9118#	9128#	9138#
	9154#	9155#	9160#	9161#	9165#	9176#	9185#	9201#	9202#	9207#	9208#	9212#	9236#	9279#	
M$IOSE	2431#	4807#													
M$LDR0	2771#	4807#	5652#	5655#	5658#	5662#	5675#	5770#	7235#	7239#	7260#	7264#	7283#	7298#	7320#
	7336#														
M$MASK	2390#	4807#													
M$MCHI	87#	4807#													
M$MCLO	2327#	4807#													
M$MSK1	2402#	4807#													
M$POP	2646#	4807#	4861#	4909#	4926#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#
	5566#	5568#	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#
	5583#	5584#	5585#	5586#	5587#	5588#	5589#	5594#	5624#	5753#	5774#	5787#	5804#	5820#	5857#
	5871#	5908#	5922#	5923#	5952#	5953#	5979#	5987#	5989#	6008#	6015#	6017#	6036#	6043#	6045#
	6074#	6075#	6105#	6120#	6121#	6145#	6161#	6162#	6188#	6200#	6201#	6222#	6231#	6232#	6261#
	6290#	6310#	6311#	6339#	6359#	6360#	6388#	6408#	6409#	6437#	6457#	6458#	6490#	6514#	6515#
	6549#	6575#	6576#	6604#	6624#	6625#	6653#	6673#	6674#	6702#	6722#	6723#	6751#	6771#	6772#
	6800#	6820#	6821#	6849#	6869#	6870#	6898#	6918#	6919#	6947#	6967#	6968#	7005#	7024#	7025#
	7053#	7092#	7093#	7128#	7154#	7159#	7160#	7198#	7219#	7220#	7246#	7271#	7308#	7346#	7381#
	7415#	7448#	7498#	7542#	7587#	7619#	7672#	7706#	7707#	7751#	7756#	7797#	7802#	7843#	7848#
	7889#	7894#	7935#	7940#	7981#	7986#	8027#	8032#	8073#	8078#	8119#	8125#	8167#	8172#	8215#
	8221#	8263#	8268#	8310#	8315#	8357#	8362#	8404#	8409#	8452#	8457#	8499#	8504#	8548#	8553#
	8595#	8600#	8642#	8647#	8689#	8694#	8736#	8741#	8783#	8788#	8830#	8835#	8877#	8882#	8924#
	8929#	8972#	8977#	9019#	9024#	9066#	9071#	9113#	9118#	9160#	9165#	9207#	9212#	9247#	9282#
	9290#														
M$PRIN	2349#	4807#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#	5568#	5570#
	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#	5584#	5585#
	5586#	5587#	5588#	5589#	5592#	5593#									
M$PUSH	2337#	4807#	4813#	4857#	4894#	4923#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#
	5565#	5566#	5568#	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#
	5582#	5583#	5584#	5585#	5586#	5587#	5588#	5589#	5591#	5607#	5638#	5756#	5784#	5801#	5819#
	5838#	5863#	5891#	5896#	5911#	5931#	5935#	5968#	5970#	5980#	5997#	5999#	6009#	6025#	6027#
	6037#	6052#	6056#	6089#	6094#	6107#	6129#	6134#	6147#	6170#	6174#	6189#	6209#	6213#	6224#
	6245#	6269#	6275#	6293#	6319#	6324#	6342#	6368#	6373#	6391#	6417#	6422#	6440#	6466#	6471#
	6493#	6523#	6528#	6552#	6584#	6589#	6607#	6633#	6638#	6656#	6682#	6687#	6705#	6731#	6736#
	6754#	6780#	6785#	6803#	6829#	6834#	6852#	6878#	6883#	6901#	6927#	6932#	6950#	6976#	6981#
	7008#	7033#	7038#	7056#	7101#	7106#	7107#	7130#	7168#	7173#	7201#	7227#	7253#	7279#	7316#
	7353#	7388#	7422#	7456#	7510#	7551#	7595#	7627#	7683#	7690#	7720#	7729#	7766#	7775#	7812#
	7821#	7858#	7867#	7904#	7913#	7950#	7959#	7996#	8005#	8042#	8051#	8088#	8097#	8136#	8145#
	8183#	8193#	8232#	8241#	8279#	8288#	8326#	8335#	8373#	8382#	8421#	8430#	8468#	8477#	8517#
	8526#	8564#	8573#	8611#	8620#	8658#	8667#	8705#	8714#	8752#	8761#	8799#	8808#	8846#	8855#
	8893#	8902#	8941#	8950#	8988#	8997#	9035#	9044#	9082#	9091#	9128#	9138#	9176#	9185#	9236#
	9279#														
M$PUT	2819#	4807#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#	5568#	5570#
	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#	5584#	5585#
	5586#	5587#	5588#	5589#	5592#	5593#									
M$PUT1	2842#	4807#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#	5568#	5570#
	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#	5584#	5585#

```
            5586#   5587#   5588#   5589#   5592#   5593#
M$RADI      3151#   4807#   9238#   9239#   9240#   9241#
M$RBRO      2787#   4807#
M$RNRO      2802#   4807#   5675#
M$SETS      3071#   4807#   4813#   4857#   4894#   4923#   5556#   5557#   5558#   5559#   5560#   5561#   5562#   5563#   5564#
            5565#   5566#   5568#   5570#   5571#   5572#   5573#   5574#   5575#   5576#   5577#   5578#   5579#   5580#   5581#
            5582#   5583#   5584#   5585#   5586#   5587#   5588#   5589#   5591#   5607#   5638#   5756#   5784#   5801#   5819#
            5838#   5863#   5891#   5896#   5911#   5931#   5935#   5968#   5970#   5980#   5997#   5999#   6009#   6025#   6027#
            6037#   6052#   6056#   6089#   6094#   6107#   6129#   6134#   6147#   6170#   6174#   6189#   6209#   6213#   6224#
            6245#   6269#   6275#   6293#   6319#   6324#   6342#   6368#   6373#   6391#   6417#   6422#   6440#   6466#   6471#
            6493#   6523#   6528#   6552#   6584#   6589#   6607#   6633#   6638#   6656#   6682#   6687#   6705#   6731#   6736#
            6754#   6780#   6785#   6803#   6829#   6834#   6852#   6878#   6883#   6901#   6927#   6932#   6950#   6976#   6981#
            7008#   7033#   7038#   7056#   7101#   7106#   7107#   7130#   7168#   7173#   7201#   7227#   7253#   7279#   7316#
            7353#   7388#   7422#   7456#   7510#   7551#   7595#   7627#   7683#   7690#   7720#   7729#   7766#   7775#   7812#
            7821#   7858#   7867#   7904#   7913#   7950#   7959#   7996#   8005#   8042#   8051#   8088#   8097#   8136#   8145#
            8183#   8193#   8232#   8241#   8279#   8288#   8326#   8335#   8373#   8382#   8421#   8430#   8468#   8477#   8517#
            8526#   8564#   8573#   8611#   8620#   8658#   8667#   8705#   8714#   8752#   8761#   8799#   8808#   8846#   8855#
            8893#   8902#   8941#   8950#   8988#   8997#   9035#   9044#   9082#   9091#   9128#   9138#   9176#   9185#   9236#
            9279#
M$STAR      2468#   4807#
M$SVC       2746#   4807#   5383    5556#   5557#   5558#   5559#   5560#   5561#   5562#   5563#   5564#   5565#   5566#   5568#
            5570#   5571#   5572#   5573#   5574#   5575#   5576#   5577#   5578#   5579#   5580#   5581#   5582#   5583#   5584#
            5585#   5586#   5587#   5588#   5589#   5592#   5593#   5594#   5615#   5624#   5652#   5655#   5658#   5662#   5675#
            5753#   5770#   5774#   5785#   5787#   5803#   5804#   5820#   5845#   5852    5856#   5857#   5869    5871#   5896#
            5902    5903#   5908#   5911#   5916    5917#   5922#   5923#   5935#   5942    5943#   5948    5949#   5952#   5953#
            5970#   5977    5978#   5979#   5980#   5985    5986#   5987#   5989#   5999#   6006    6007#   6008#   6009#   6014
            6015#   6017#   6027#   6034    6035#   6036#   6037#   6042    6043#   6045#   6056#   6064    6065#   6070    6071#
            6074#   6075#   6094#   6100    6101#   6105#   6107#   6114    6115#   6120#   6121#   6134#   6140    6141#   6145#
            6147#   6155    6156#   6161#   6162#   6174#   6180    6181#   6188#   6189#   6194    6195#   6200#   6201#   6213#
            6220    6221#   6222#   6224#   6229    6230#   6231#   6232#   6257    6259#   6261#   6275#   6285    6286#   6290#
            6293#   6304    6305#   6310#   6311#   6324#   6334    6335#   6339#   6342#   6353    6354#   6359#   6360#   6373#
            6383    6384#   6388#   6391#   6402    6403#   6408#   6409#   6422#   6432    6433#   6437#   6440#   6451    6452#
            6457#   6458#   6471#   6484    6485#   6490#   6493#   6507    6508#   6514#   6515#   6528#   6542    6544#   6549#
            6552#   6568    6569#   6575#   6576#   6589#   6599    6600#   6604#   6607#   6618    6619#   6624#   6625#   6638#
            6648    6649#   6653#   6656#   6667    6668#   6673#   6674#   6687#   6697    6698#   6702#   6707    6716    6717#
            6722#   6723#   6736#   6746    6747#   6751#   6754#   6765    6766#   6771#   6772#   6785#   6795    6796#   6800#
            6803#   6814    6815#   6820#   6821#   6834#   6844    6845#   6849#   6852#   6863    6864#   6869#   6870#   6883#
            6893    6894#   6898#   6901#   6912    6913#   6918#   6919#   6932#   6942    6943#   6947#   6950#   6961    6962#
            6967#   6968#   6981#   6999    7000#   7005#   7008#   7017    7018#   7024#   7025#   7038#   7048    7049#   7053#
            7056#   7068    7069#   7084    7085#   7092#   7093#   7106#   7107#   7123    7124#   7128#   7130#   7148    7149#
            7154#   7159#   7160#   7173#   7192    7193#   7198#   7201#   7213    7214#   7219#   7220#   7229#   7235#   7239#
            7241    7243    7246#   7260#   7264#   7266    7268    7271#   7283#   7298#   7304#   7305    7308#   7320#   7336#
            7338    7342    7346#   7354#   7376    7377#   7378#   7381#   7410    7411#   7412#   7415#   7443    7444#   7445#
            7448#   7496    7498#   7537    7542#   7574    7584    7587#   7617    7619#   7653    7654#   7668#   7672#   7690#
            7704    7705#   7706#   7707#   7729#   7745    7746#   7751#   7752#   7756#   7775#   7791    7792#   7797#   7798#
            7802#   7821#   7837    7838#   7843#   7844#   7848#   7867#   7883    7884#   7889#   7890#   7894#   7913#   7929
            7930#   7935#   7936#   7940#   7959#   7975    7976#   7981#   7982#   7986#   8005#   8021    8022#   8027#   8028#
            8032#   8051#   8067    8068#   8073#   8074#   8078#   8097#   8113    8114#   8119#   8120#   8125#   8145#   8161
            8162#   8167#   8168#   8172#   8193#   8209    8210#   8215#   8216#   8221#   8241#   8257    8258#   8263#   8264#
            8268#   8288#   8304    8305#   8310#   8311#   8315#   8335#   8351    8352#   8357#   8358#   8362#   8382#   8398
            8399#   8404#   8405#   8409#   8430#   8446    8447#   8452#   8453#   8457#   8477#   8493    8494#   8499#   8500#
            8504#   8526#   8542    8543#   8548#   8549#   8553#   8573#   8589    8590#   8595#   8596#   8600#   8620#   8636
            8637#   8642#   8643#   8647#   8667#   8683    8684#   8689#   8690#   8694#   8714#   8730    8731#   8736#   8737#
            8741#   8761#   8777    8778#   8783#   8784#   8788#   8808#   8824    8825#   8830#   8831#   8835#   8855#   8871
            8872#   8877#   8878#   8882#   8902#   8918    8919#   8924#   8925#   8929#   8950#   8966    8967#   8972#   8973#
            8977#   8997#   9013    9014#   9019#   9020#   9024#   9044#   9060    9061#   9066#   9067#   9071#   9091#   9107
```

	9108# 9212#	9113#	9114#	9118#	9138#	9154	9155#	9160#	9161#	9165#	9185#	9201	9202#	9207#	9208#
M$TLAB	2739#	4807#	5383#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#	5568#
	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#	5584#
	5585#	5586#	5587#	5588#	5589#	5592#	5593#	5594#	5624#	5652#	5655#	5658#	5662#	5675#	5753#
	5770#	5774#	5785#	5787#	5803#	5804#	5820#	5845#	5852#	5856#	5857#	5869#	5871#	5896#	5902#
	5903#	5908#	5911#	5916#	5917#	5922#	5923#	5935#	5942#	5943#	5948#	5949#	5952#	5953#	5970#
	5977#	5978#	5979#	5980#	5985#	5986#	5987#	5989#	5999#	6006#	6007#	6008#	6009#	6014#	6015#
	6017#	6027#	6034#	6035#	6036#	6037#	6042#	6043#	6045#	6056#	6064#	6065#	6070#	6071#	6074#
	6075#	6094#	6100#	6101#	6105#	6107#	6114#	6115#	6120#	6121#	6134#	6140#	6141#	6145#	6147#
	6155#	6156#	6161#	6162#	6174#	6180#	6181#	6188#	6189#	6194#	6195#	6200#	6201#	6213#	6220#
	6221#	6222#	6224#	6229#	6230#	6231#	6232#	6257#	6259#	6261#	6275#	6285#	6286#	6290#	6293#
	6304#	6305#	6310#	6311#	6324#	6334#	6335#	6339#	6342#	6353#	6354#	6359#	6360#	6373#	6383#
	6384#	6388#	6391#	6402#	6403#	6408#	6409#	6422#	6432#	6433#	6437#	6440#	6451#	6452#	6457#
	6458#	6471#	6484#	6485#	6490#	6493#	6507#	6508#	6514#	6515#	6528#	6542#	6544#	6549#	6552#
	6568#	6569#	6575#	6576#	6589#	6599#	6600#	6604#	6607#	6618#	6619#	6624#	6625#	6638#	6648#
	6649#	6653#	6656#	6667#	6668#	6673#	6674#	6687#	6697#	6698#	6702#	6705#	6716#	6717#	6722#
	6723#	6736#	6746#	6747#	6751#	6754#	6765#	6766#	6771#	6772#	6785#	6795#	6796#	6800#	6803#
	6814#	6815#	6820#	6821#	6834#	6844#	6845#	6849#	6852#	6863#	6864#	6869#	6870#	6883#	6893#
	6894#	6898#	6901#	6912#	6913#	6918#	6919#	6932#	6942#	6943#	6947#	6950#	6961#	6962#	6967#
	6968#	6981#	6999#	7000#	7005#	7008#	7017#	7018#	7024#	7025#	7038#	7048#	7049#	7053#	7056#
	7068#	7069#	7084#	7085#	7092#	7093#	7106#	7107#	7123#	7124#	7128#	7130#	7148#	7149#	7154#
	7159#	7160#	7173#	7192#	7193#	7198#	7201#	7213#	7214#	7219#	7220#	7229#	7235#	7239#	7241#
	7243#	7246#	7260#	7264#	7266#	7268#	7271#	7283#	7298#	7304#	7305#	7308#	7320#	7336#	7338#
	7342#	7346#	7354#	7376#	7377#	7378#	7381#	7410#	7411#	7412#	7415#	7443#	7444#	7445#	7448#
	7496#	7498#	7537#	7542#	7574#	7584#	7587#	7617#	7619#	7653#	7654#	7668#	7672#	7690#	7704#
	7705#	7706#	7707#	7729#	7745#	7746#	7751#	7752#	7756#	7775#	7791#	7792#	7797#	7798#	7802#
	7821#	7837#	7838#	7843#	7844#	7848#	7867#	7883#	7884#	7889#	7890#	7894#	7913#	7929#	7930#
	7935#	7936#	7940#	7959#	7975#	7976#	7981#	7982#	7986#	8005#	8021#	8022#	8027#	8028#	8032#
	8051#	8067#	8068#	8073#	8074#	8078#	8097#	8113#	8114#	8119#	8120#	8125#	8145#	8161#	8162#
	8167#	8168#	8172#	8193#	8209#	8210#	8215#	8216#	8221#	8241#	8257#	8258#	8263#	8264#	8268#
	8288#	8304#	8305#	8310#	8311#	8315#	8335#	8351#	8352#	8357#	8358#	8362#	8382#	8398#	8399#
	8404#	8405#	8409#	8430#	8446#	8447#	8452#	8453#	8457#	8477#	8493#	8494#	8499#	8500#	8504#
	8526#	8542#	8543#	8548#	8549#	8553#	8573#	8589#	8590#	8595#	8596#	8600#	8620#	8636#	8637#
	8642#	8643#	8647#	8667#	8683#	8684#	8689#	8690#	8694#	8714#	8730#	8731#	8736#	8737#	8741#
	8761#	8777#	8778#	8783#	8784#	8788#	8808#	8824#	8825#	8830#	8831#	8835#	8855#	8871#	8872#
	8877#	8878#	8882#	8902#	8918#	8919#	8924#	8925#	8929#	8950#	8966#	8967#	8972#	8973#	8977#
	8997#	9013#	9014#	9019#	9020#	9024#	9044#	9060#	9061#	9066#	9067#	9071#	9091#	9107#	9108#
	9113#	9114#	9118#	9138#	9154#	9155#	9160#	9161#	9165#	9185#	9201#	9202#	9207#	9208#	9212#
M$TSTL	2728#	4807#	5383#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#	5568#
	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#	5584#
	5585#	5586#	5587#	5588#	5589#	5592#	5593#	5594#	5624#	5652#	5655#	5658#	5662#	5675#	5753#
	5770#	5774#	5785#	5787#	5803#	5804#	5820#	5845#	5852#	5856#	5857#	5869#	5871#	5896#	5902#
	5903#	5908#	5911#	5916#	5917#	5922#	5923#	5935#	5942#	5943#	5948#	5949#	5952#	5953#	5970#
	5977#	5978#	5979#	5980#	5985#	5986#	5987#	5989#	5999#	6006#	6007#	6008#	6009#	6014#	6015#
	6017#	6027#	6034#	6035#	6036#	6037#	6042#	6043#	6045#	5056#	6064#	6065#	6070#	6071#	6074#
	6075#	6094#	6100#	6101#	6105#	6107#	6114#	6115#	6120#	6121#	6134#	6140#	6141#	6145#	6147#
	6155#	6156#	6161#	6162#	6174#	6180#	6181#	6188#	6189#	6194#	6195#	6200#	6201#	6213#	6220#
	6221#	6222#	6224#	6229#	6230#	6231#	6232#	6257#	6259#	6261#	6275#	6285#	6286#	6290#	6293#
	6304#	6305#	6310#	6311#	6324#	6334#	6335#	6339#	6342#	6353#	6354#	6359#	6360#	6373#	6383#
	6384#	6388#	6391#	6402#	6403#	6408#	6409#	6422#	6432#	6433#	6437#	6440#	6451#	6452#	6457#
	6458#	6471#	6484#	6485#	6490#	6493#	6507#	6508#	6514#	6515#	6528#	6542#	6544#	6549#	6552#
	6568#	6569#	6575#	6576#	6589#	6599#	6600#	6604#	6607#	6618#	6619#	6624#	6625#	6638#	6648#
	6649#	6653#	6656#	6667#	6668#	6673#	6674#	6687#	6697#	6698#	6702#	6705#	6716#	6717#	6722#
	6723#	6736#	6746#	6747#	6751#	6754#	6765#	6766#	6771#	6772#	6785#	6795#	6796#	6800#	6803#
	6814#	6815#	6820#	6821#	6834#	6844#	6845#	6849#	6852#	6863#	6864#	6869#	6870#	6883#	6893#

```
        6894#  6898#  6901#  6912#  6913#  6918#  6919#  6932#  6942#  6943#  6947#  6950#  6961#  6962#  6967#
        6968#  6981#  6999#  7000#  7005#  7008#  7017#  7018#  7024#  7025#  7038#  7048#  7049#  7053#  7056#
        7068#  7069#  7084#  7085#  7092#  7093#  7106#  7107#  7123#  7124#  7128#  7130#  7148#  7149#  7154#
        7159#  7160#  7173#  7192#  7193#  7198#  7201#  7213#  7214#  7219#  7220#  7229#  7235#  7239#  7241#
        7243#  7246#  7260#  7264#  7266#  7268#  7271#  7283#  7298#  7304#  7305#  7308#  7320#  7336#  7338#
        7342#  7346#  7354#  7376#  7377#  7378#  7381#  7410#  7411#  7412#  7415#  7443#  7444#  7445#  7448#
        7496#  7498#  7537#  7542#  7574#  7584#  7587#  7617#  7619#  7653#  7654#  7668#  7672#  7690#  7704#
        7705#  7706#  7707#  7729#  7745#  7746#  7751#  7752#  7756#  7775#  7791#  7792#  7797#  7798#  7802#
        7821#  7837#  7838#  7843#  7844#  7848#  7867#  7883#  7884#  7889#  7890#  7894#  7913#  7929#  7930#
        7935#  7936#  7940#  7959#  7975#  7976#  7981#  7982#  7986#  8005#  8021#  8022#  8027#  8028#  8032#
        8051#  8067#  8068#  8073#  8074#  8078#  8097#  8113#  8114#  8119#  8120#  8125#  8145#  8161#  8162#
        8167#  8168#  8172#  8193#  8209#  8210#  8215#  8216#  8221#  8241#  8257#  8258#  8263#  8264#  8268#
        8288#  8304#  8305#  8310#  8311#  8315#  8335#  8351#  8352#  8357#  8358#  8362#  8382#  8398#  8399#
        8404#  8405#  8409#  8430#  8446#  8447#  8452#  8453#  8457#  8477#  8493#  8494#  8499#  8500#  8504#
        8526#  8542#  8543#  8548#  8549#  8553#  8573#  8589#  8590#  8595#  8596#  8600#  8620#  8636#  8637#
        8642#  8643#  8647#  8667#  8683#  8684#  8689#  8690#  8694#  8714#  8730#  8731#  8736#  8737#  8741#
        8761#  8777#  8778#  8783#  8784#  8788#  8808#  8824#  8825#  8830#  8831#  8835#  8855#  8871#  8872#
        8877#  8878#  8882#  8902#  8918#  8919#  8924#  8925#  8929#  8950#  8966#  8967#  8972#  8973#  8977#
        8997#  9013#  9014#  9019#  9020#  9024#  9044#  9060#  9061#  9066#  9067#  9071#  9091#  9107#  9108#
        9113#  9114#  9118#  9138#  9154#  9155#  9160#  9161#  9165#  9185#  9201#  9202#  9207#  9208#  9212#
M$WORD  2888#  4807#  4848#  4871#  5383#  5615#  5852#  5869#  5902#  5916#  5942#  5948#  5977#  5985#  6006#
        6014#  6034#  6042#  6064#  6070#  6100#  6114#  6140#  6155#  6180#  6194#  6220#  6229#  6257#  6285#
        6304#  6334#  6353#  6383#  6402#  6432#  6451#  6484#  6507#  6542#  6568#  6599#  6618#  6648#  6667#
        6697#  6716#  6746#  6765#  6795#  6814#  6844#  6863#  6893#  6912#  6942#  6961#  6999#  7017#  7048#
        7068#  7084#  7123#  7148#  7192#  7213#  7241#  7243#  7266#  7268#  7305#  7338#  7342#  7376#  7378#
        7410#  7412#  7443#  7445#  7496#  7537#  7574#  7584#  7617#  7653#  7668#  7704#  7745#  7752#  7791#
        7798#  7837#  7844#  7883#  7890#  7929#  7936#  7975#  7982#  8021#  8028#  8067#  8074#  8113#  8120#
        8161#  8168#  8209#  8216#  8257#  8264#  8304#  8311#  8351#  8358#  8398#  8405#  8446#  8453#  8493#
        8500#  8542#  8549#  8589#  8596#  8636#  8643#  8683#  8690#  8730#  8737#  8777#  8784#  8824#  8831#
        8871#  8878#  8918#  8925#  8966#  8973#  9013#  9020#  9060#  9067#  9107#  9114#  9154#  9161#  9201#
        9208#  9238#  9239#  9240#  9241#  9298
M$XFER  2410#  4807#
OPEN    1694#  4807#
POINTF  1702#  4807#  4838
PRINTB  1768#  4807#  5556  5557  5558  5559  5560  5561  5562  5563  5564  5565  5566  5568  5570
        5571   5572   5573  5574  5575  5576  5577  5578  5579  5580  5581  5582  5583  5584  5585
        5586   5587   5588  5589
PRINTF  1811#  4807#  5592  5593
PRINTS  1854#  4807#
PRINTX  1897#  4807#
READBL  1940#  4807#
READEF  1949#  4807#  5652  5655  5658  5662
RERROR  5155#  7123   7148  7192  7213
RFLAGS  1967#  4807#
ROMCLK  5194#  5230   5232  5234  5240  5247  5254  5261  5269  5283  5344  5362  5364  5367  5372
        5376   5403   5414  5416  5424  5426  6278  6280  6297  6299  6327  6329  6346  6348  6376
        6378   6395   6397  6425  6427  6444  6446  6475  6477  6498  6500  6532  6534  6557  6559
        6592   6594   6611  6613  6641  6643  6660  6662  6690  6692  6709  6711  6739  6741  6758
        6760   6788   6790  6807  6809  6837  6839  6856  6858  6886  6888  6905  6907  6935  6937
        6954   6956   6986  6994  7012  7041  7043  7060  7062  7079  7111  7115  7117  7136  7140
        7142   7178   7183  7186  7205  7207  7237  7262  7296  7334  7366  7369  7399  7403  7405
        7433   7436   7467  7489  7491  7528  7530  7561  7564  7567  7578  7604  7607  7610  7646
        7692   7694   7696  7698  7733  7737  7739  7779  7783  7785  7825  7829  7831  7871  7875
        7877   7917   7921  7923  7963  7967  7969  8009  8013  8015  8055  8059  8061  8101  8105
        8107   8149   8153  8155  8197  8201  8203  8245  8249  8251  8292  8296  8298  8339  8343
        8345   8386   8390  8392  8434  8438  8440  8481  8485  8487  8530  8534  8536  8577  8581
```

```
        8583    8624    8628    8630    8671    8675    8677    8718    8722    8724    8765    8769    8771    8812    8816
        8818    8859    8863    8865    8906    8910    8912    8954    8958    8960    9001    9005    9007    9048    9052
        9054    9095    9099    9101    9142    9146    9148    9189    9193    9195
SETPRI  1977#   4807#   7235    7239    7260    7264    7283    7298    7320    7336
SETVEC  1986#   4807#
SLASH   1998#   4807#
STARS   2015#   4807#
SVC     2036#   4806#   4807
XFER    2299#   4807#   5615#   7378#   7412#   7445#   7668#   7752#   7798#   7844#   7890#   7936#   7982#   8028#   8074#
        8120#   8168#   8216#   8264#   8311#   8358#   8405#   8453#   8500#   8549#   8596#   8643#   8690#   8737#   8784#
        8831#   8878#   8925#   8973#   9020#   9067#   9114#   9161#   9208#
XFERF   2310#   4807#
XFERT   2319#   4807#
$MD     5540#   5556    5557    5558    5559    5560    5561    5562    5563    5564    5565    5566    5568    5570    5571
        5572    5573    5574    5575    5576    5577    5578    5579    5580    5581    5582    5583    5584    5585    5586
        5587    5588    5589


. ABS.  040004      000


 ERRORS DETECTED:  0

 SAIL:CZDMPA,CZDMPA/CRF/NL:TOC=CZDMP.MLB,CZDMPA.P11
 RUN-TIME: 192 224 23 SECONDS
 RUN-TIME RATIO: 2338/441=5.2
 CORE USED:  21K  (41 PAGES)
```