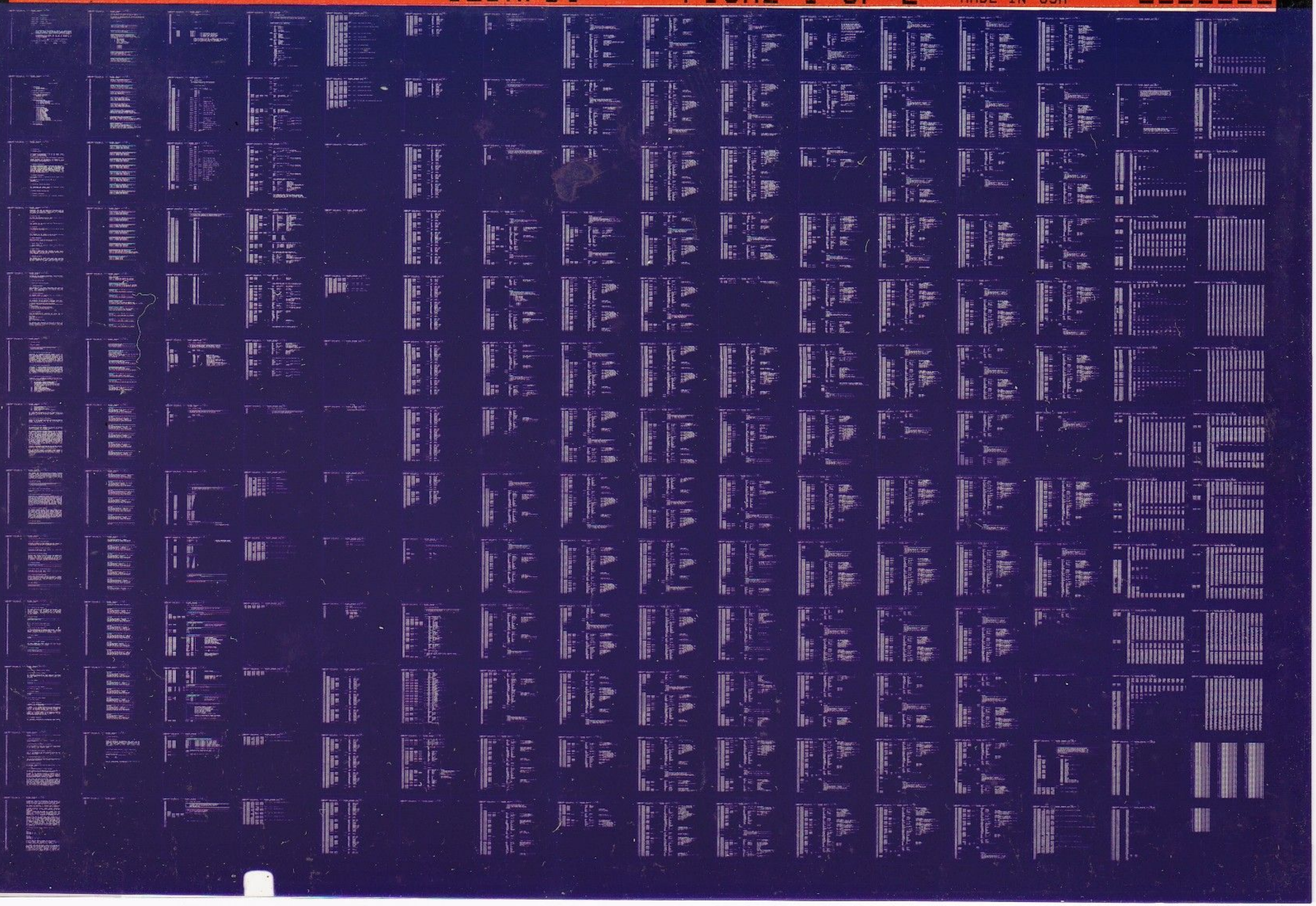
DMP11,DMR11, M8207 STATIC DIAG#1 AH-E226D-MC
M8207 CZDMPD0 FICHE 1 OF 2



DMP11,DMR11, M8207 STATIC DIAG#1 AH-E226D-MC JAN 1983
M8207 CZDMPD0 FICHE 2 OF 2

```
.REM @

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

PRODUCT CODE:   AC-E225D-MC

PRODUCT NAME:   CZDMPDO M8207 STATIC DIAG #1

PRODUCT DATE:   JANUARY  1983

MAINTAINER:     DIAGNOSTICS MERRIMACK

AUTHOR:         ED BADGER
```

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 PROGRAM ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST OUT THE M8200, M8204, OR M8207 MICROBPROCESSOR. 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 'LSO.

2.0 HARDWARE REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE M8207 STATIC LOCIC 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

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 ERROR LOG. THIS LOG MAY BE PRINTED BY USING THE "PRINT" COMMAND (SEE SECTION 6.3.8).

5.0 PROGRAM LOAD MEDIA

THIS PROGRAM CAN BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER OR FROM ACT, SLIDE, OR APT SYSTEMS, OR FROM ANY MEDIA SUPPORTED BY XXDP+. WHEN USING THE PAPER TAPE ABSOLUTE LOADER, THE PROGRAM SHOULD BE LOADED FIRST,

FOLLOWED BY THE DIAGNOSTIC SUPERVISOR. WHEN USING XXDP+, THE DIAGNOSTIC SUPERVISOR SHOULD BE LOADED FIRST, FOLLOWED BY THE DIAGNOSTIC PROGRAM.

6.0 OPERATING INSTRUCTIONS

6.1 LOADING AND STARTING PROCEDURES

6.1.1 LOADING PROCEDURES

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

6.1.2 STARTING PROCEDURES

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

6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

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

A) LOAD AND START DIAGNOSTIC USING RUN COMMAND
B) RECEIVE DIAGNOSTIC SUPERVISOR 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-D-0
M8207 DIAG.#1 OF 2
UNIT IS M8200,M8204,OR M8207
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:

HOE	HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED
LOE	LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR
IER	INHIBIT ERROR REPORTING
IBE	INHIBIT BASIC ERROR REPORTS
IXE	INHIBIT EXTENDED ERROR REPORTS
PRI	DIRECT ALL MESSAGES TO A LINE PRINTER
PNT	PRINT NUMBER OF TEST BEING EXECUTED
BOE	BELL ON ERROR

```
311                                   UAM   RUN IN UNATTENDED MODE, BYPASSING MANUAL
312                                         INTERVENTION TESTS
313                                   ISR   INHIBIT STATISTICAL REPORTS
314                                   IDU   INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
315                                   LOT   LOOP ON TEST
316
317                                 THE  FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0
318                                 ARE CLEARED.  A FLAG NOT SPECIFIED IS CLEARED.  IF THE FLAGS
319                                 SWITCH  IS  NOT GIVEN ALL FLAGS ARE CLEARED.  SEE EXAMPLE AT
320                                 END OF 6.3.1.5.
321
322
323                                 6.3.1.4 END OF PASS SWITCH (/EOP:<INCR>)
324
325                                 <INCR> IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF
326                                 PASSES)  IT  IS  DESIRED  THAT  THE  END  OF PASS MESSAGE BE
327                                 PRINTED.  THE DEFAULT IS AT THE  END  OF  EVERY  PASS.   SEE
328                                 EXAMPLE AT END OF 6.3.1.5.
329
330
331                                 6.3.1.5 EFFECT OF START COMMAND
332
333                                 THE  EFFECT OF THE START COMMAND IS TO INITIATE THE HARDWARE
334                                 PARAMETER DIALOGUE, THE  SOFTWARE  PARAMETER  DIALOGUE,  AND
335                                 THEN THE DIAGNOSTIC TESTS THEMSELVES.
336
337                                 THE  HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION
338                                 "# UNITS?" TO WHICH THE  OPERATOR  REPLIES  WITH  A  DECIMAL
339                                 NUMBER  N FROM 1 TO 16. THE TERM "UNIT" REFERS TO THE DEVICE
340                                 TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED.  FOLLOWING
341                                 THIS  ARE THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES WILL
342                                 BE BUILT.  EACH P-TABLE IS A CORE-RESIDENT TABLE  CONTAINING
343                                 ALL  THE  HARDWARE  INFORMATION  FOR ONE UNIT.  THE OPERATOR
344                                 MUST SUPPLY N (NUMBER OF UNITS) VALUES  FOR  EACH  QUESTION.
345                                 HE  MAY  DO  THIS  BY GIVING ONE ANSWER TO EACH QUESTION (IN
346                                 WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR
347                                 BY  GIVING  N  VALUES, SEPARATED BY COMMAS, TO EACH QUESTION
348                                 (SERIES WILL BE POSED ONCE).  EACH QUESTION IS  FOLLOWED  BY
349                                 THE  RESPONSE  RADIX  (D  FOR  DECIMAL,  B FOR BINARY, O FOR
350                                 OCTAL, L FOR YES/NO) IN PARENTHESES AND  THE  DEFAULT  VALUE
351                                 AFTER THE PARENTHESES.
352
353                                 FOLLOWING  THE HARDWARE QUESTIONS ARE THE SOFTWARE QUESTIONS
354                                 TO BUILD THE SOFTWARE TABLES, WHICH DEFINE THE  MODE  (QUICK
355                                 VERIFY ETC.) THAT THE DIAGNOSTIC WILL EXECUTE IN.
356
357                                 WHEN  THE QUESTION "# UNITS?" IS ANSWERED, MEMORY STORAGE IS
358                                 ALLOCATED FOR THE P-TABLES, AND IF THERE IS  NOT  ENOUGH  TO
359                                 ACCOMMODATE THEM THE MESSAGE "TOO MANY UNITS" IS ISSUED.  IN
360                                 THIS CASE THE DIAGNOSTIC MUST BE EXECUTED MORE THAN ONCE  TO
361                                 TEST ALL UNITS.
362
363                                 EXAMPLE:
364
365                                 STA/TESTS:1:2-4:6:8-10/PASS:3/FLAGS:IER:HOE=1:UAM:LOE
366
```

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

6.3.2 RESTART COMMAND

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

6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

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

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

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

6.3.2.3 EFFECT OF RESTART COMMAND

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

6.3.3 CONTINUE COMMAND

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

****

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

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

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

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

6.3.3.3 EFFECT OF CONTINUE COMMAND

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

6.3.4 PROCEED COMMAND

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

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

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

6.3.4.2 EFFECT OF PROCEED COMMAND

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

6.3.5 ADD COMMAND

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

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

479
480 <UNIT-LIST> IS AS IN THE RESTART COMMAND.
481
482
483 6.3.5.2 EFFECT OF ADD COMMAND
484
485 THE UNITS SPECIFIED ARE ADDED TO THE TEST SEQUENCE. EACH
486 UNIT MUST HAVE A P-TABLE IN MEMORY DUE TO AN EARLIER
487 HARDWARE DIALOGUE. THIS COMMAND MUST BE FOLLOWED BY A
488 RESTART OR CONTINUE. THE UNITS SWITCH MUST BE SPECIFIED.
489 THE ADD COMMAND IS MEANINGFUL ONLY FOR UNITS THAT WERE
490 PREVIOUSLY DROPPED.
491
492
493 6.3.6 DROP COMMAND
494
495 **********
496 DRO(P)/UNITS:<UNIT-LIST>
497 **********
498
499
500 6.3.6.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)
501
502 <UNIT-LIST> IS AS IN THE RESTART COMMAND.
503
504
505 6.3.6.2 EFFECT OF DROP COMMAND
506
507 THE UNITS SPECIFIED WILL BE DROPPED FROM TESTING. THE UNITS
508 WILL BE RESELECTED ONLY BY THE EXECUTION OF AN ADD OR START
509 COMMAND. THE UNITS SWITCH MUST BE ENTERED. THIS COMMAND
510 MUST BE FOLLOWED BY A RESTART OR A CONTINUE COMMAND.
511
512
513 6.3.7 PRINT COMMAND
514
515 **********
516 PRI(NT)
517 **********
518
519
520 6.3.7.1 EFFECT OF PRINT COMMAND
521
522 THE TOTAL NUMBER OF ERRORS FOR EACH UNIT SINCE THE LAST
523 START OR RESTART COMMAND ARE PRINTED. THE ISR (INHIBIT
524 STATISTICAL REPORTING) FLAG IS CLEARED.
525
526
527 6.3.8 DISPLAY COMMAND
528
529 **********
530 DIS(PLAY)/UNITS:<UNIT-LIST>
531 **********
532
533
534 6.3.8.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

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

5.3.8.2 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 C'EARED.

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: (O) 7?

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

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

THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SELO) 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 000-770 (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).


                        7.0 TEST DESCRIPTION



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


                        ********************* TEST 2 *********************
                        *VERIFY THAT RUN CAN BE CLEARED
                        ************************************************

                        ********************* TEST 3 *********************
                        *UNIBUS REGISTER WORD DUAL ADDRESSING TEST
                        *LOAD ALL REGISTERS WITH INCREMENTING PATTERN
                        *READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING
                        *THE SEQUENCE:
                        *       1.  CLEAR REGISTER
                        *       2.  WRITE PATTERN
                        *       3.  VERIFY PATTERN
                        *       4.  DO ALL 4 REGISTERS
                        *       5.  READ ALL BACK IF ERRORS,
                        *           DUAL ADDRESS PROBLEM.
                        *
                        *           1 IN REG 0
                        *           2 IN REG 2
                        *           3 IN REG 4
                        *           4 IN REG 6
                        ************************************************


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


                        ********************* TEST 5 *********************
                        *CONTROL STATUS REGISTER WRITE/READ TEST
                        *SET BIT9, VERIFY BIT9 WAS SET
                        *CLEAR BIT9, VERIFY BIT9 WAS CLEARED
                        ************************************************


                        ********************* TEST 6 *********************
```

```
                        *CONTROL STATUS REGISTER WRITE/READ TEST
                        *SET BIT11, VERIFY BIT11 WAS SET
                        *CLEAR BIT11, VERIFY BIT11 WAS CLEARED
                        ************************************************


                        ******************** TEST 7 ********************
                        *CONTROL STATUS REGISTER WRITE/READ TEST
                        *SET BIT12, VERIFY BIT12 WAS SET
                        *CLEAR BIT 12, VERIFY BIT 12 WAS CLEARED
                        ************************************************


                        ******************** TEST 8 ********************
                        *CONTROL OUT REGISTER WRITE/READ TEST
                        *FLOAT A ONE THROUGH SEL2
                        ************************************************


                        ******************** TEST 9 ********************
                        *PORT4 REGISTER WRITE/READ TEST
                        *FLOAT A ONE THROUGH PORT4 REGISTER
                        *FLOAT A ZERO THROUGH PORT4 REGISTER
                        ************************************************


                        ******************** TEST 10 *******************
                        *PORT6 REGISTER WRITE/READ TEST
                        *FLOAT A ONE THROUGH PORT6 REGISTER
                        *FLOAT A ZERO THROUGH PORT6 REGISTER
                        ************************************************


                        ******************** TEST 11 *******************
                        *UNIBUS REGISTER BYTE DUAL ADDRESSING TEST
                        *LOAD ALL REGISTERS WITH INCREMENTING PATTERN
                        *READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING
                        ************************************************


                        ******************** TEST 12 *******************
                        *MAINTENANCE INSTRUCTION REGISTER TEST
                        *VERIFY THAT THE MAINT IR CAN BE WRITTEN TO ALL ZEROS'
                        *AND ALL ONES'.  VERIFY THAT IS IS CLEARED ON A BUS RESET.
                        ************************************************


                        ******************** TEST 13 *******************
                        *MICRO PROCESSOR TEST
                        *LOAD KP 06 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
                        ************************************************
```

```
                        ********** TEST 14 **********
                        *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
                        *FLOAT A 1 THROUGH IBUS* REGISTER 0
                        *FLOAT A 0 THROUGH IBUS* REGISTER 0
                        ************************************************


                        ********** TEST 15 **********
                        *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
                        *FLOAT A 1 THROUGH IBUS* REGISTER 2
                        *FLOAT A 0 THROUGH IBUS* REGISTER 2
                        ************************************************


                        ********** TEST 16 **********
                        *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
                        *FLOAT A 1 THROUGH IBUS* REGISTER 4
                        *FLOAT A 0 THROUGH IBUS* REGISTER 4
                        ************************************************


                        ********** TEST 17 **********
                        *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
                        *FLOAT A 1 THROUGH IBUS* REGISTER 5
                        *FLOAT A 0 THROUGH IBUS* REGISTER 5
                        ************************************************


                        ********** TEST 18 **********
                        *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
                        *FLOAT A 1 THROUGH IBUS* REGISTER 10
                        *FLOAT A 0 THROUGH IBUS* REGISTER 10
                        ************************************************


                        ********** TEST 19 **********
                        *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
                        *FLOAT A 1 THROUGH IBUS* REGISTER 11
                        *FLOAT A 0 THROUGH IBUS* REGISTER 11
                        ************************************************


                        ********** TEST 20 **********
                        *MICRO PROCESSOR IBUS REBISTER WRITE/READ TEST
                        *FLOAT A 1 THROUGH IBUS REGISTER 0
                        *FLOAT A 0 THROUGH IBUS REGISTER 0
                        ************************************************


                        ********** TEST 21 **********
                        *MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
                        *FLOAT A 1 THROUGH IBUS REGISTER 1
                        *FLOAT A 0 THROUGH IBUS REGISTER 1
                        ************************************************
```

```
                        ********** TEST 22 **********
                        *MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
                        *FLOAT A 1 THROUGH IBUS REGISTER 2
                        *FLOAT A 0 THROUGH IBUS REGISTER 2
                        ************************************************


                        ********** TEST 23 **********
                        *MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
                        *FLOAT A 1 THROUGH IBUS REGISTER 3
                        *FLOAT A 0 THROUGH IBUS REGISTER 3
                        ************************************************


                        ********** TEST 24 **********
                        *MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
                        *FLOAT A 1 THROUGH IBUS REGISTER 4
                        *FLOAT A 0 THROUGH IBUS REGISTER 4
                        ************************************************


                        ********** TEST 25 **********
                        *MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
                        *FLOAT A 1 THROUGH IBUS REGISTER 5
                        *FLOAT A 0 THROUGH IBUS REGISTER 5
                        ************************************************


                        ********** TEST 26 **********
                        *MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
                        *FLOAT A 1 THROUGH IBUS REGISTER 6
                        *FLOAT A 0 THROUGH IBUS REGISTER 6
                        ************************************************


                        ********** TEST 27 **********
                        *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
                        *FLOAT A 1 THROUGH IBUS* REGISTER 7
                        *FLOAT A 0 THROUGH IBUS* REGISTER 7
                        ************************************************


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


                        ********** TEST 29 **********
                        *MICRO PROCESSOR BR REGISTER TEST
                        *FLOAT A 1 THROUGH THE BR
                        *FLOAT A 0 THROUGH THE BR
                        ************************************************
```

```
                        ;***** ********** TEST 30 ******************
                        ;*SCRATCH PAD TEST
                        ;*FLOAT A 1 THROUGH EACH SCRATCH PAD LOCATION
                        ;*FLOAT A 0 THROUGH EACH SCRATCH PAD LOCATION
                        ;*************************************************


                        ;********************** TEST 31 ******************
                        ;*SCRATCH PAD DUAL ADDRESSING TEST
                        ;*WRITE AN INCREMENTING PATTERN IN ALL SP LOCATIONS
                        ;*READ ALL SP LOCATIONS TO VERIFY CORRECT ADDRESSING
                        ;*************************************************


                        ;********************** TEST 32 ******************
                        ;*INTERRUPT TEST
                        ;*TEST THAT DEVICE CAN INTERRUPT TO VECTOR A
                        ;*************************************************


                        ;********************** TEST 33 ******************
                        ;*INTERRUPT TEST
                        ;*TEST THAT DEVICE CAN INTERRUPT TO VECTOR B
                        ;*************************************************


                        ;********************** TEST 34 ******************
                        ;*PRIORITY INTERRUPT TEST
                        ;*SET PS TO ALL BR LEVELS EQUAL OR GREATER THAN
                        ;*THE M8200,4,7 LEVEL, VERIFY THAT M8200,4,7 DOES NOT INTERRUPT
                        ;*************************************************


                        ;********************** TEST 35 ******************
                        ;*PRIORITY INTERRUPT TESTS
                        ;*SET PS TO ALL BR LEVELS LESS THAN THE M8200,4,7 LEVEL
                        ;*VERIFY THAT ALL M8200,4,7 WILL INTERRUPT
                        ;*************************************************


                        ;********************** TEST 36 ******************
                        ;*NPR TEST
                        ;*TEST OF DATO, 1 WORD FROM UPROC TO 11 MEMORY
                        ;*************************************************


                        ;********************** TEST 37 ******************
                        ;*NPR TEST
                        ;*TEST OF DATI, 1 WORD FROM 11 MEMORY TO UPROC
                        ;*************************************************


                        ;********************** TEST 38 ******************
                        ;*NPR TEST
                        ;*TEST OF DATOB, 1 BYTE FROM UPROC TO 11 MEMORY
                        ;*************************************************
```

```
                        ********** TEST 39 **********
                        *TEST OF EA BITS 16 AND 17
                        *DO A DATO TO AN ADDRESS USING OUT BA BITS 16 AND 17
                        *VERIFY CORRECT RESULTS
                        ************************************************


                        ********** TEST 40 **********
                        *TEST OF EA BITS 16 AND 17
                        *DO A DATI USING IN BA BITS 16 AND 17
                        *VERIFY CORRECT RESULTS
                        *IN ORDER TO DO THIS TEST, WE WILL READ THE DATA FROM THE
                        *CONSOL TTY CSR IF ONE EXISTS
                        *IF NO COSOL TTY CSR AT ADDRESS 177560, THIS TEST
                        *WILL BE SKIPPED
                        ************************************************


                        ********** TEST 41 **********
                        *NPR NON-EXISTENT MEMORY TEST
                        *DO A DATO TO A NON -EXISTENT ADDRESS
                        *VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
                        ************************************************


                        ********** TEST 42 **********
                        *NPR NON-EXISTENT MEMORY TEST
                        *DO A DATI FROM A NON-EXISTENT ADDRESS
                        *VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
                        ************************************************


                        ********** TEST 43 **********
                        *NPR TEST
                        *USING DATO, NPR A BINARY COUNT (0-377)
                        *FROM MICRO-PROCESSOR TO ALL AVAILABLE MEMORY
                        ************************************************


                        ********** TEST 44 **********
                        *ALU C BIT TEST
                        *TEST THAT AN ADD OF 377 AND 377 WILL SET THE C BIT
                        ************************************************


                        ********** TEST 45 **********
                        *ALU TEST
                        *TEST OF ALU FUNCTION SEL B WITH C BIT CLEARED
                        *ALU FUNCTION (B)    CODE=11
                        *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                        *PERFORM THE FUNCTION, VERIFY THE RESULTS
                        ************************************************
```

```
1039                            ********** TEST 46 **********
1040                            *ALU TEST
1041                            *TEST OF ALU FUNCTION SEL A WITH C BIT CLEARED
1042                            *ALU FUNCTION (A)    CODE=10
1043                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1044                            *PERFORM THE FUNCTION, VERIFY THE RESULTS
1045                            ************************************************
1046
1047
1048                            ********** TEST 47 **********
1049                            *ALU TEST
1050                            *TEST OF ALU FUNCTION A OR NOTB WITH C BIT CLEARED
1051                            *ALU FUNCTION (A OR NOTB)    CODE=12
1052                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1053                            *PERFORM THE FUNCTION, VERIFY THE RESULTS
1054                            ************************************************
1055
1056
1057                            ********** TEST 48 **********
1058                            *ALU TEST
1059                            *TEST OF ALU FUNCTION A AND B  WITH C BIT CLEARED
1060                            *ALU FUNCTION (A AND B)    CODE=13
1061                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1062                            *PERFORM THE FUNCTION, VERIFY THE RESULTS
1063                            ************************************************
1064
1065
1066                            ********** TEST 49 **********
1067                            *ALU TEST
1068                            *TEST OF ALU FUNCTION A OR B WITH C BIT CLEARED
1069                            *ALU FUNCTION  (A OR B)    CODE=14
1070                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1071                            *PERFORM THE FUNCTION, VERIFY THE RESULTS
1072                            ************************************************
1073
1074
1075
1076                            ********** TEST 50 **********
1077                            *ALU TEST
1078                            *TEST OF ALU FUNCTION A XOR B WITH C BIT
1079                            *ALU FUNCTION (A XOR B)    CODE=15
1080                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1081                            *PERFORM THE FUNCTION, VERIFY THE RESULTS
1082                            ************************************************
1083
1084
1085                            ********** TEST 51 **********
1086                            *ALU TEST
1087                            *TEST OF ALU FUNCTION ADD WITH C BIT CLEARED
1088                            *ALU FUNCTION (A PLUS B)    CODE=00
1089                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1090                            *PERFORM THE FUNCTION, VERIFY THE RESULTS
1091                            ************************************************
1092
1093
1094                            ********** TEST 52 **********
```

```
1095                            *ALU TEST
1096                            *TEST OF ALU FUNCTION 2A W.C WITH C BIT CLEARED
1097                            *ALU FUNCTION (A PLUS A PLUS C)    CODE=6
1098                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1099                            *PERFORM THE FUNCTION, VERIFY THE RESULTS
1100                            ************************************************
1101
1102
1103                            ******************** TEST 53 **********************
1104                            *ALU TEST
1105                            *TEST OF ALU FUNCTION SUB WITH C BIT CLEARED
1106                            *ALU FUNCTION (A-B)    CODE=16
1107                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1108                            *PERFORM THE FUNCTION, VERIFY THE RESULTS
1109                            ************************************************
1110
1111
1112                            ******************** TEST 54 **********************
1113                            *ALU TEST
1114                            *TEST OF ALU FUNCTION ADD W/C WITH C BIT CLEARED
1115                            *ALU FUNCTION (A PLUS B PLUS C)    CODE=01
1116                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1117                            *PERFORM THE FUNCTION, VERIFY THE RESULTS
1118                            ************************************************
1119
1120
1121                            ******************** TEST 55 **********************
1122                            *ALU TEST
1123                            *TEST OF ALU FUNCTION SUB W/C WITH C BIT CLEARED
1124                            *ALU FUNCTION (A-B-C)    CODE=2
1125                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1126                            *PERFORM THE FUNCTION. VERIFY THE RESULTS
1127                            ************************************************
1128
1129
1130                            ******************** TEST 56 **********************
1131                            *ALU TEST
1132                            *TEST OF ALU FUNCTION INC A WITH C BIT CLEARED
1133                            *ALU FUNCTION (A PLUS 1)    CODE =3
1134                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1135                            *PERFORM THE FUNCTION, VERIFY THE RESULTS
1136                            ************************************************
1137
1138
1139                            ******************** TEST 57 **********************
1140                            *ALU TEST
1141                            *TEST OF ALU FUNCTION 2A WITH C BIT CLEARED
1142                            *ALU FUNCTION (A PLUS A)    CODE=5
1143                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1144                            *PERFORM THE FUNCTION, VERIFY THE RESULTS
1145                            ************************************************
1146
1147
1148                            ******************** TEST 58 **********************
1149                            *ALU TEST
1150                            *TEST OF ALU FUNCTION A PLUS C WITH C BIT CLEARED
```

```
1151                            *ALU FUNCTION (A PLUS C)    CODE=4
1152                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1153                            *PERFORM THE FUNCTION, VERIFY THE RESULTS
1154                            ************************************************
1155
1156
1157                            ******************** TEST 59 **********************
1158                            *ALU TEST
1159                            *TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT CLEARED
1160                            *ALU FUNCTION (A-B-1)    CODE=17
1161                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1162                            *PERFORM THE FUNCTION, VERIFY THE RESULTS
1163                            ************************************************
1164
1165
1166                            ******************** TEST 60 **********************
1167                            *ALU TEST
1168                            *TEST OF ALU FUNCTION DEC A WITH C BIT CLEARED
1169                            *ALU FUNCTION (A-1)    CODE=7
1170                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1171                            *PERFORM THE FUNCTION, VERIFY THE RESULTS
1172                            ************************************************
1173
1174
1175                            ******************** TEST 61 **********************
1176                            *ALU TEST
1177                            *TEST OF ALU FUNCTION SEL B WITH C BIT SET
1178                            *ALU FUNCTION (B)    CODE=11
1179                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1180                            *PERFORM THE FUNCTION, VERIFY THE RESULTS
1181                            ************************************************
1182
1183
1184                            ******************** TEST 62 **********************
1185                            *ALU TEST
1186                            *TEST OF ALU FUNCTION SEL A WITH C BIT SET
1187                            *ALU FUNCTION (A)    CODE=10
1188                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1189                            *PERFORM THE FUNCTION, VERIFY THE RESULTS
1190                            ************************************************
1191
1192
1193                            ******************** TEST 63 **********************
1194                            *ALU TEST
1195                            *TEST OF ALU FUNCTION A OR NOTB WITH C BIT SET
1196                            *ALU FUNCTION (A OR NOTB)    CODE=12
1197                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1198                            *PERFORM THE FUNCTION, VERIFY THE RESULTS
1199                            ************************************************
1200
1201
1202                            ******************** TEST 64 **********************
1203                            *ALU TEST
1204                            *TEST OF ALU FUNCTION A AND B WITH C BIT SET
1205                            *ALU FUNCTION (A AND B)    CODE=13
1206                            *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
```

```
1207                                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
1208                                    ************************************************
1209
1210
1211                                    ******************** TEST 65 ********************
1212                                    *ALU TEST
1213                                    *TEST OF ALU FUNCTION A OR B WITH C BIT SET
1214                                    *ALU FUNCTION (A OR B)    CODE=14
1215                                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1216                                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
1217                                    ************************************************
1218
1219
1220                                    ******************** TEST 66 ********************
1221                                    *ALU TEST
1222                                    *TEST OF ALU FUNCTION A XOR B WITH C BIT SET
1223                                    *ALU FUNCTION (A XOR B)    CODE=15
1224                                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1225                                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
1226                                    ************************************************
1227
1228
1229                                    ******************** TEST 67 ********************
1230                                    *ALU TEST
1231                                    *TEST OF ALU FUNCTION ADD WITH C BIT SET
1232                                    *ALU FUNCTION (A PLUS B)     CODE=00
1233                                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1234                                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
1235                                    ************************************************
1236
1237
1238                                    ******************** TEST 68 ********************
1239                                    *ALU TEST
1240                                    *TEST OF ALU FUNCTION 2A W/C WITH C BIT SET
1241                                    *ALU FUNCTION (A PLUS A PLUS C)    CODE=6
1242                                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1243                                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
1244                                    ************************************************
1245
1246
1247                                    ******************** TEST 69 ********************
1248                                    *ALU TEST
1249                                    *TEST OF ALU FUNCTION SUB WITH C BIT SET
1250                                    *ALU FUNCTION (A-B)    CODE=16
1251                                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1252                                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
1253                                    ************************************************
1254
1255
1256                                    ******************** TEST 70 ********************
1257                                    *ALU TEST
1258                                    *TEST OF ALU FUNCTION ADD W/C WITH C BIT SET
1259                                    *ALU FUNCTION (A PLUS B PLUS C)    CODE=01
1260                                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1261                                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
1262                                    ************************************************
```

```
1263
1264
1265                    ********** TEST 71 **********
1266                    *ALU TEST
1267                    *TEST OF ALU FUNCTION SUB W/C WITH C BIT SET
1268                    *ALU FUNCTION (A-B-C)    CODE=2
1269                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1270                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
1271                    ************************************************
1272
1273
1274                    ********** TEST 72 **********
1275                    *ALU TEST
1276                    *TEST OF ALU FUNCTION INC A WITH C BIT SET
1277                    *ALU FUNCTION (A PLUS 1)    CODE=3
1278                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1279                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
1280                    ************************************************
1281
1282
1283                    ********** TEST 73 **********
1284                    *ALU TEST
1285                    *TEST OF ALU FUNCTION 2A WITH C BIT SET
1286                    *ALU FUNCTION (A PLUS A)    CODE=5
1287                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1288                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
1289                    ************************************************
1290
1291
1292                    ********** TEST 74 **********
1293                    *ALU TEST
1294                    *TEST OF ALU FUNCTION A PLUS C WITH C BIT SET
1295                    *ALU FUNCTION (A PLUS C)    CODE=4
1296                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1297                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
1298                    ************************************************
1299
1300
1301                    ********** TEST 75 **********
1302                    *ALU TEST
1303                    *TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT SET
1304                    *ALU FUNCTION (A-B-1)    CODE=17
1305                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1306                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
1307                    ************************************************
1308
1309
1310                    ********** TEST 76 **********
1311                    *ALU TEST
1312                    *TEST OF ALU FUNCTION DEC A WITH C BIT SET
1313                    *ALU FUNCTION (A-1)    CODE=7
1314                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
1315                    *PERFORM THE FUNCTION, VERIFY THE RESULT
1316                    ************************************************
1317
1318
```

8.0 ERROR INFORMATION

8.1 ERROR REPORTING

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

THE FOLLOWING EXAMPLE PROVIDES A TYPICAL ERROR REPORT.

```
CZDMP DVC FTL ERR 00003 TST 029 SUB 000 PC:022626

BR REGISTER DATA TEST
UNIT=00; FAILING UNIT ADDRESS=160170

GOOD            BAD
177776          000011
```

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

@

```
1353
1354
1355
1356
1357
```

```
1358                                    .TITLE CZDMPD0 M8207 STATIC DIAG #1
1359            002000                  .=2000
1360
1361
1362
1363
1364
1365
1366                                    .MCALL  SVC
1367    002000                          SVC                             ; INITIALIZE SUPERVISOR MACROS
1368
1369
1370
1371
1372
1373    002000                          BGNMOD  CZDMP
1374
1375
1376            000000                  $LSTIN= 0
1377            000000                  $LSTTAG= 0
1378            000000                  SVCINS= 0         LIST INSTRUCTIONS, SHIFTED RIGHT
1379            000000                  SVCTST= 0       ; LIST TEST TAGS, SHIFTED RIGHT
1380            000000                  SVCSUB= 0       ; LIST SUBTEST TAGS, SHIFTED RIGHT
1381            000000                  SVCGBL= 0       ; LIST GLOBAL TAGS, SHIFTED RIGHT
1382            000000                  SVCTAG= 0       ; LIST OTHER TAGS, SHIFTED RIGHT
1383
1384                                    ;     CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
1385                                    ;     TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS.  CHANGE THE
1386                                    ;     SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS.  YOU MAY
1387                                    ;     CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
1388
1389
1390                                    .ENABL  AMA
```

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

002000                                  POINTER BGNAU,BGNDU


                                        HEADER  CZDMP,D,0,120.,0
002000                          L$NAME::                ;DIAGNOSTIC NAME
002000  103                             .ASCII /C/
002001  132                             .ASCII /Z/
002002  104                             .ASCII /D/
002003  115                             .ASCII /M/
002004  120                             .ASCII /P/
002005  000                             .BYTE   0
002006  000                             .BYTE   0
002007  000                             .BYTE   0
002010                          L$REV::                 ;REVISION LEVEL
002010  104                             .ASCII  /D/
002011                          L$DEPO::                ;0
002011  060                             .ASCII  /0/
002012                          L$UNIT::                ;NUMBER OF UNITS
002012  000000                          .WORD   0
002014                          L$TIML::                ;LONGEST TEST TIME
002014  000170                          .WORD   120.
002016                          L$HPCP::                ;POINTER TO H.W. QUES.
002016  034640                          .WORD   L$HARD
002020                          L$SPCP::                ;POINTER TO S.W. QUES.
002020  000000                          .WORD   0
002022                          L$HPTP::                ;PTR. TO DEF. H.W. PTABLE
002022  002364                          .WORD   L$HW
002024                          L$SPTP::                ;PTR. TO S.W. PTABLE
002024  000000                          .WORD   0
002026                          L$LADP::                ;DIAG. END ADDRESS
002026  040004                          .WORD   L$LAST
002030                          L$STA::                 ;RESERVED FOR APT STATS
002030  000000                          .WORD   0
002032                          L$CO::
002032  000000                          .WORD   0
002034                          L$DTYP::                ;DIAGNOSTIC TYPE
002034  000000                          .WORD   0
002036                          L$APT::                 ;APT EXPANSION
002036  000000                          .WORD   0
002040                          L$DTP::                 ;PTR. TO DISPATCH TABLE
002040  002132                          .WORD   L$DISPATCH
002042                          L$PRIO::                ;DIAGNOSTIC RUN PRIORITY
002042  000000                          .WORD   0
002044                          L$ENVI::                ;FLAGS DESCRIBE HOW IT WAS SETUP
002044  000000                          .WORD   0
002046                          L$EXP1::                ;EXPANSION WORD
002046  000000                          .WORD   0
002050                          L$MREV::                ;SVC REV AND EDIT #
002050  003                             .BYTE   C$REVISION
002051  003                             .BYTE   C$EDIT
```

```
1447  002052                          L$EF::                  ;DIAG. EVENT FLAGS
1448  002052  000000                          .WORD   0
1449  002054  000000                          .WORD   0
1450  002056                          L$SPC::
1451  002056  000000                          .WORD   0
1452  002060                          L$DEVP::                ; POINTER TO DEVICE TYPE LIST
1453  002060  003130                          .WORD   L$DVTYP
1454  002062                          L$REPP::                ;PTR. TO REPORT CODE
1455  002062  000000                          .WORD   0
1456  002064                          L$EXP4::
1457  002064  000000                          .WORD   0
1458  002066                          L$EXP5::
1459  002066  000000                          .WORD   0
1460  002070                          L$AUT::                 ;PTR. TO ADD UNIT CODE
1461  002070  011364                          .WORD   L$AU
1462  002072                          L$DUT::                 ;PTR. TO DROP UNIT CODE
1463  002072  011360                          .WORD   L$DU
1464  002074                          L$LUN::                 ;LUN FOR EXERCISERS TO FILL
1465  002074  000000                          .WORD   0
1466  002076                          L$DESP::                ;POINTER TO DIAG. DESCRIPTION
1467  002076  002414                          .WORD   L$DESC
1468  002100                          L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
1469  002100  104035                          EMT     E$LOAD
1470  002102                          L$ETP::                 ;POINTER TO ERRTBL
1471  002102  000000                          .WORD   0
1472  002104                          L$ICP::                 ;PTR. TO INIT CODE
1473  002104  010570                          .WORD   L$INIT
1474  002106                          L$CCP::                 ;PTR. TO CLEAN-UP CODE
1475  002106  011354                          .WORD   L$CLEAN
1476  002110                          L$ACP::                 ;PTR. TO AUTO CODE
1477  002110  011256                          .WORD   L$AUTO
1478  002112                          L$PRT::                 ;PTR. TO PROTECT TABLE
1479  002112  002122                          .WORD   L$PROT
1480  002114                          L$TEST::                ;TEST NUMBER
1481  002114  000000                          .WORD   0
1482  002116                          L$DLY::                 ;DELAY COUNT
1483  002116  000000                          .WORD   0
1484  002120                          L$HIME::                ;PTR. TO HIGH MEM
1485  002120  000000                          .WORD   0
1486
1487
1488
1489  002122                                  BGNPROT
1490  002122                          L$PROT::
1491  002122  177777                          .WORD   -1
1492  002124  177777                          .WORD   -1
1493  002126  177777                          .WORD   -1
1494  002130                                  ENDPROT
1495
```

```
                                .SBTTL  DISPATCH TABLE

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

002130                                  DISPATCH 76.
002130  000114                          .WORD   76
002132                          L$DISPATCH::
002132  011366                          .WORD   T1
002134  011514                          .WORD   T2
002136  011560                          .WORD   T3
002140  011746                          .WORD   T4
002142  012112                          .WORD   T5
002144  012246                          .WORD   T6
002146  012376                          .WORD   T7
002150  012526                          .WORD   T8
002152  012670                          .WORD   T9
002154  013054                          .WORD   T10
002156  013240                          .WORD   T11
002160  013426                          .WORD   T12
002162  013576                          .WORD   T13
002164  013704                          .WORD   T14
002166  014134                          .WORD   T15
002170  014364                          .WORD   T16
002172  014614                          .WORD   T17
002174  015044                          .WORD   T18
002176  015340                          .WORD   T19
002200  015634                          .WORD   T20
002202  016064                          .WORD   T21
002204  016314                          .WORD   T22
002206  016544                          .WORD   T23
002210  016774                          .WORD   T24
002212  017224                          .WORD   T25
002214  017454                          .WORD   T26
002216  017704                          .WORD   T27
002220  020134                          .WORD   T28
002222  020432                          .WORD   T29
002224  020662                          .WORD   T30
002226  021226                          .WORD   T31
002230  021540                          .WORD   T32
002232  021702                          .WORD   T33
002234  022044                          .WORD   T34
002236  022220                          .WORD   T35
002240  022424                          .WORD   T36
002242  022570                          .WORD   T37
002244  022740                          .WORD   T38
002246  023104                          .WORD   T39
002250  023272                          .WORD   T40
002252  023506                          .WORD   T41
002254  023724                          .WORD   T42
002256  024056                          .WORD   T43
002260  024274                          .WORD   T44
002262  024434                          .WORD   T45
002264  024640                          .WORD   T46
```

```
1552  002266  025044                          .WORD   T47
1553  002270  025250                          .WORD   T48
1554  002272  025454                          .WORD   T49
1555  002274  025660                          .WORD   T50
1556  002276  026064                          .WORD   T51
1557  002300  026270                          .WORD   T52
1558  002302  026474                          .WORD   T53
1559  002304  026702                          .WORD   T54
1560  002306  027106                          .WORD   T55
1561  002310  027312                          .WORD   T56
1562  002312  027516                          .WORD   T57
1563  002314  027722                          .WORD   T58
1564  002316  030126                          .WORD   T59
1565  002320  030332                          .WORD   T60
1566  002322  030536                          .WORD   T61
1567  002324  030742                          .WORD   T62
1568  002326  031146                          .WORD   T63
1569  002330  031352                          .WORD   T64
1570  002332  031556                          .WORD   T65
1571  002334  031762                          .WORD   T66
1572  002336  032166                          .WORD   T67
1573  002340  032372                          .WORD   T68
1574  002342  032576                          .WORD   T69
1575  002344  033002                          .WORD   T70
1576  002346  033206                          .WORD   T71
1577  002350  033412                          .WORD   T72
1578  002352  033616                          .WORD   T73
1579  002354  034022                          .WORD   T74
1580  002356  034226                          .WORD   T75
1581  002360  034432                          .WORD   T76
1582                                          ;LNT.ED DIFINED AT END OF PROGRAM TO BE LAST TEST NUMBER.
1583
1584
1585
1586
1587
1588
```

```
                                .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
002362  000013                        .WORD   L10001-L$HW/2
002364                          L$HW::
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
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
002412  000000                          .WORD   L10002-L$SW/2
002414                          L$SW::
002414                          SFPTBL::


002414                                  ENDSW
002414                          L10002:
```

```
                                .SBTTL  GLOBAL EQUATES SECTION

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

002414                                  EQUALS
                                ;
                                ; BIT DIFINITIONS
                                ;
100000                          BIT15== 100000
040000                          BIT14== 40000
020000                          BIT13== 20000
010000                          BIT12== 10000
004000                          BIT11== 4000
002000                          BIT10== 2000
001000                          BIT09== 1000
000400                          BIT08== 400
000200                          BIT07== 200
000100                          BIT06== 100
000040                          BIT05== 40
000020                          BIT04== 20
000010                          BIT03== 10
000004                          BIT02== 4
000002                          BIT01== 2
000001                          BIT00== 1
                                ;
001000                          BIT9==  BIT09
000400                          BIT8==  BIT08
000200                          BIT7==  BIT07
000100                          BIT6==  BIT06
000040                          BIT5==  BIT05
000020                          BIT4==  BIT04
000010                          BIT3==  BIT03
000004                          BIT2==  BIT02
000002                          BIT1==  BIT01
000001                          BIT0==  BIT00
                                ;
                                ; EVENT FLAG DEFINITIONS
                                ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                                ;
000040                          EF.START==      32.                             ; START COMMAND WAS ISSUED
000037                          EF.RESTART==    31.                             ; RESTART COMMAND WAS ISSUED
```

```
1699    000036          EF.CONTINUE==   30.                      ; CONTINUE COMMAND WAS ISSUED
1700    000035          EF.NEW==        29.                      ; A NEW PASS HAS BEEN STARTED
1701    000034          EF.PWR==        28.                      ; A POWER-FAIL/POWER-UP OCCURRED
1702                    ;
1703                    ;
1704                    ; PRIORITY LEVEL DEFINITIONS
1705                    ;
1706    000340          PRI07== 340
1707    000300          PRI06== 300
1708    000240          PRI05== 240
1709    000200          PRI04== 200
1710    000140          PRI03== 140
1711    000100          PRI02== 100
1712    000040          PRI01== 40
1713    000000          PRI00== 0
1714                    ;
1715                    ;OPERATOR FLAG BITS
1716                    ;
1717    000004          EVL==       4
1718    000010          LOT==      10
1719    000020          ADR==      20
1720    000040          IDU==      40
1721    000100          ISR==     100
1722    000200          UAM==     200
1723    000400          BOE==     400
1724    001000          PNT==    1000
1725    002000          PRI==    2000
1726    004000          IXE==    4000
1727    010000          IBE==   10000
1728    020000          IER==   20000
1729    040000          LOE==   40000
1730    100000          HOE==  100000
1731
1732
1733
1734
1735
1736                    ;*****************************************************************************
1737                    ;* INSTRUCTION DEFINITIONS
1738                    ;*****************************************************************************
1739    022626          POP2SP=22626            ;INCREMENT STACK TWICE
1740
1741
1742
1743                    ;;*****************************************************************************
1744                    ;* PROGRAM EVENT FLAG DEFINITIONS
1745                    ;;*****************************************************************************
1746
1747
1748
```

```
                                .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                  LOCDEV: .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 NPR'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
```

```
1805  002620  000000                  FLAG:   .WORD   0               ;SCRATCH STORAGE
1806  002622  000000                  RUN:    .WORD   0               ;POINTER TO RUNNING DEVICES
1807  002624  000000                  MR0:    .WORD   0
1808  002626  000000                  WTYPE:  .WORD   0
1809  002630  000000                  TYPE:   .WORD   0
1810  002632  000000                  $GDADR: .WORD   0               ;CONTAINS ADDRESS OF 'GOOD' DATA
1811  002634  000000                  $BDADR: .WORD   0               ;CONTAINS ADDRESS OF 'BAD' DATA
1812  002636  000000                  $GDDAT: .WORD   0               ;CONTAINS 'GOOD' DATA
1813  002640  000000                  $BDDAT: .WORD   0               ;CONTAINS 'BAD' DATA
1814  002642  000000                          .WORD   0               ;RESERVED--NOT TO BE USED
1815  002644  000000                          .WORD   0
1816  002646  000000                  FTIME:  .WORD   0
1817  002650  000000                  SAVE4:  .WORD   0
1818  002652  000000                  SAVE6:  .WORD   0
1819
1820                                  ;******************************************************************************
1821                                  ;* DATA PATTERNS
1822                                  ;******************************************************************************
1823  002654      000     377     000 MEMDAT: .BYTE   0,-1,0,-1,125,252,125,252
1824  002657      377     125     252
1825  002662      125     252
1826  002664      000     000     377 SPDAT:  .BYTE   0,0,-1,-1,125,125,252,252
1827  002667      377     125     125
1828  002672      252     252
1829                                          .EVEN
1830
1831                                  ;******************************************************************************
1832                                  ;* PROGRAM CONTROL FLAGS
1833                                  ;******************************************************************************
1834  002674      000                 INIFLG: .BYTE   0               ;PROGRAM INITIALIZING FLAG
1835          002676                          .EVEN
1836  002676      000                 LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
1837  002677      000                 QV.FLG: .BYTE   0               ;QUICK VERIFY FLAG
1838                                          .EVEN
1839
1840                                  ;******************************************************************************
1841                                  ;* DEFINITION OF M8200,4,7 STATUS WORDS - STAT1,STAT2,STAT3
1842                                  ;*
1843                                  ;*
1844                                  ;*      STAT1 - BITS 00-08 IS M8200,4,7 VECTOR ADDRESS
1845                                  ;*              BIT15=1 LINE UNIT IS AN M8203
1846                                  ;*              BIT14=0 NO TEST CONNECTOR(S) USED
1847                                  ;*              BIT14=1 H-XXX TEST CONNECTOR WILL BE USED
1848                                  ;*              BIT13=0 LINE UNIT IS AN M8201
1849                                  ;*              BIT13=1 LINE UNIT IS AN M8202
1850                                  ;*              BIT12=1 NO LINE UNIT
1851                                  ;*              BITS 09-11 IS M8200,4,7 PRIORITY LEVEL
1852                                  ;*
1853                                  ;*      STAT2 - LOW BYTE IS SWITCH PACK #1 (DDCMP LINE NUMBER)
1854                                  ;*              HIGH BYTE IS SWITCH PACK #2 (BM873 BOOT ADDRESS)
1855                                  ;*
1856                                  ;*      STAT3 - BIT0=1 DO FREE RUNNING TESTS ON M8200,4,7
1857                                  ;*
1858                                  ;******************************************************************************
1859  002700  000000                  STAT1:  .WORD   0
1860  002702  000000                  STAT2:  .WORD   0
```

```
1861  002704  000000                  STAT3:  .WORD   0
1862
1863                                  ;*********************************************************************
1864                                  ;* POINTERS TO M8200,4,7 VECTORS AND REGISTERS
1865                                  ;*********************************************************************
1866  002706  000000                  KMRVEC: 0       ;POINTER TO M8200,4,7 RCV INTRPT VECTOR
1867  002710  000000                  KMRLVL: 0       ;POINTER TO M8200,4,7 RCV INTRPT SERVICE PS
1868  002712  000000                  KMTVEC: 0       ;POINTER TO M8200,4,7 TX INTRPT VECTOR
1869  002714  000000                  KMTLVL: 0       ;POINTER TO M8200,4,7 TX INTRPT SERVICE PS
1870  002716  000000                  KMCSR:  0       ;POINTER TO M8200,4,7 CONTROL STATUS REGISTER
1871  002720  000000                  KMCSRH: 0       ;POINTER TO M8200,4,7 CONTROL STATUS REGISTER HIGH BYTE
1872  002722  000000                  KMCTL:  0       ;POINTER TO M8200,4,7 CONTROL OUT REGISTER
1873  002724  000000                  KMP04:  0
1874  002726  000000                  KMP06:  0       ;POINTER TO M8200,4,7 PORT REGISTER - SEL6
1875
1876                                  ;;****  PRIMARY REG ADRS STORAGE FOR THIS UNIT  ******
1877                                  ;THESE LOCATIONS WILL BE LOADED FOR THE CURRENT UNIT, IN INIT CODE
1878  002730                          REGADR:
1879
1880                                  ;;****  STACK USED FOR SUBROUTINE LINKAGE  *****
1881  002730  000100                          .BLKW   100
1882  003130                          SSTACK:
1883
1884
1885
1886
1887
1888
1889
```

```
                                        .SBTTL  GLOBAL TEXT SECTION

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

                                ;;*************************************************************************
                                ;* NAMES OF DEVICES SUPPORTED BY PROGRAM
                                ;;*************************************************************************
003130                                  DEVTYP  <M8200,M8204,OR M8207>
003130                          L$DVTYP::
003130  034115  030062  026060          .ASCIZ  /M8200,M8204,OR M8207/
003136  034115  030062  026064
003144  051117  046440  031070
003152  033460     000
        003156                          .EVEN


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

```
                                .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
003156                          .MSTCLR:
003156  112777  000100  177534          MOVB    #BIT6,@#MCSRH   ;SET INST.
003164  142777  000300  177526          BICB    #BIT6!BIT7,@#MCSRH
003172  000205                          RTS     R5
                                        ;
003174  000024                          .BLKW   20.             ;PATCH AREA.


003244                          ENDBUG:
                                ;       UNSAFE TO PATCH ANY OTHER AREA.


003244                          .ROMCLK:
003244  152777  000002  177446          BISB    #BIT1,@#MCSRH
003252  012577  177450                  MOV     (R5)+,@#MP06
003256  152777  000003  177434          BISB    #BIT1!BIT0,@#MCSRH
003264  142777  000007  177426          BICB    #BIT2!BIT1!BIT0,@#MCSRH
003272  000205                          RTS     R5

003274                          CLRALL:
                                        ;CLEARS C & Z BITS AND BR
003274                                  ROMCLK
003274  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
003300  000400                          400                     ;0 TO BR
003302                                  ROMCLK
003302  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
003306  063220                          63220                   ;SP(0) TO BR
003310                                  ROMCLK
003310  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
003314  060400                          60400                   ;BR,SP(0) + BR
003316  000207                          RTS     PC

003320                          SETBR0:
                                        ;SETS BR0 BIT
003320                                  ROMCLK
003320  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
003324  000401                          401                     ;1 TO BR
003326  000207                          RTS     PC
```

```
2031
2032  003330                          SETBR1:
2033                                          ;THIS SUBROUTINE SETS BR1 BIT
2034
2035  003330                          ROMCLK                  ;NEXT WORD IS INSTRUCTION
2036  003330  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
2037  003334  000402                         000402                  ;BR_002
2038  003336  000207                         RTS     PC
2039
2040  003340                          SETBR4:
2041                                          ;THIS SUBROUTINE SETS BR4 BIT
2042
2043  003340                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION
2044  003340  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
2045  003344  000402                         402
2046  003346  000207                         RTS     PC
2047
2048  003350                          SETBR7:
2049                                          ;THIS SUBROUTINE SETS BR7 BIT
2050
2051  003350                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION
2052  003350  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
2053  003354  000600                         600
2054  003356  000207                         RTS     PC
2055
2056
2057  003360                          SETZ:
2058                                          ;THIS SUBROUTINE SETS THE Z BIT
2059
2060  003360                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION
2061  003360  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
2062  003364  000777                         000777                  ;BR_377
2063  003366  000207                         RTS     PC
2064
2065  003370                          RAMDAT:
2066                                          ;THIS SUBROUTINE LOADS R4 WITH THE LOWEST
2067                                          ;8 BITS OF THE CRAM PC.
2068
2069  003370  017605  000000                 MOV     @(SP),R5        ;GOOD DATA
2070  003374  062716  000002                 ADD     #2,(SP)         ;ADJUST STACK
2071  003400  005011                         CLR     (R1)            ;CLEAR BIT10
2072  003402  052711  000400                 BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION IN CRAM THAT
2073                                                                  ;JUMPED TO, IT LOADS BR WITH IT
2074  003406  005011                         CLR     (R1)            ;CLR BIT8
2075  003410                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION
2076  003410  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
2077  003414  061225                         061225                  ;MOV BR TO PORT 5
2078  003416  116104  000005                 MOVB    5(R1),R4        ;PUT "FOUND" IN R4
2079  003422  000207                         RTS     PC              ;RETURN
2080
2081
2082  003424                          MEMSET:
2083                                          ;THIS SUBROUTINE LOADS CRAM WITH SPECIAL INSTRUCTIONS
2084                                          ;FOR THE CRAM JUMP TEST. ALL CRAM LOCATIONS ARE LOADED
2085                                          ;WITH INSTRUCTIONS THAT MOVE A 37 TO THE BR, EXCEPT THE
2086                                          ;FOLLOWING CRAM ADDRESSES: 0,1,4,7,525,1777. THESE LOCATIONS
```

```
2087                                    ;CONTAIN INSTRUCTIONS WHICH LOAD THE BR WITH THE LOWEST
2088                                    ;8 BITS OF THAT CRAM ADDRESS.
2089
2090  003424  005000                    CLR     R0              ;R0 = CRAM ADDRESS
2091  003426  012711  002000    1$:     MOV     #BIT10,(R1)     ;SET ROMO
2092  003432  010061  000004            MOV     R0,4(R1)        ;LOAD CRAM ADDRESS
2093  003436  012761  000437  000006    MOV     #437,6(R1)      ;LOAD INSTRUCTION
2094  003444  052711  020000            BIS     #BIT13,(R1)     ;WRITE INSTRUCTION IN CRAM
2095  003450  005200                    INC     R0              ;NEXT ADDRESS
2096  003452  022700  002000            CMP     #2000,R0        ;DONE YET?
2097  003456  001363                    BNE     1$              ;BR IF NO
2098  003460  005000                    CLR     R0              ;INDEX REGISTER
2099  003462  012711  002000    2$:     MOV     #BIT10,(R1)     ;SET ROMO
2100  003466  016061  003522  000004    MOV     CRAMA(R0),4(R1) ;LOAD CRAM ADDRESS IN SEL4
2101  003474  016061  003536  000006    MOV     INSTU(R0),6(R1) ;LOAD INSTRUCTION TO BE WRITTEN
2102  003502  052711  020000            BIS     #BIT13,(R1)     ;WRITE CRAM!
2103  003506  005720                    TST     (R0)+           ;NEXT
2104  003510  022700  000014            CMP     #14,R0          ;DONE YET?
2105  003514  001362                    BNE     2$              ;BR IF NO
2106  003516  005011                    CLR     (R1)            ;CLEAR ALL BITS
2107  003520  000207                    RTS     PC              ;RETURN
2108
2109  003522  000000  000001  000004  CRAMA:  .WORD   0,1,4,7,1777,525
2110  003530  000007  001777  000525
2111
2112  003536  000400            INSTU:  000400          ;BR_0
2113  003540  000401                    000401          ;BR_1
2114  003542  000404                    000404          ;BR_4
2115  003544  000407                    000407          ;BR_7
2116  003546  000777                    000777          ;BR_377
2117  003550  000525                    000525          ;BR_125
2118
2119  003552                    SETVEC:
2120                                    ;THIS SUBROUTINE LOADS THE VECTORS AND VECTOR LEVELS
2121
2122  003552  012577  177130            MOV     (R5)+,@KMRVEC   ;LOAD BASE VECTOR
2123  003556  012577  177130            MOV     (R5)+,@KMTVEC   ;LOAD VECTOR + 2
2124  003562  012577  177122            MOV     (R5)+,@KMRLVL   ;LOAD VECTOR + 4
2125  003566  012577  177122            MOV     (R5)+,@KMTLVL   ;LOAD VECTOR + 6
2126  003572  000205                    RTS     R5              ;RETURN
2127
2128
2129  003574                    NPRSET:
2130                                    ;THIS SUBROUTINE LOADS IBUS REGISTERS 0-7
2131                                    ;WITH NPR INFORMATION (INBA, OUTBA, OUT DATA)
2132
2133  003574  010246                    MOV     R2,-(SP)        ;SAVE R2
2134  003576  005002                    CLR     R2              ;START AT IBUS REG 0
2135  003600  112561  000004    1$:     MOVB    (R5)+,4(R1)     ;LOAD PORT4
2136  003604  042737  000017  003622    BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
2137  003612  050237  003622            BIS     R2,2$           ;ADD ADDRESS TO INSTRUCTION
2138  003616                            ROMCLK
2139  003616  004537  003244            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
2140  003622  122100            2$:     122100                  ;MOVE PORT4 TO IBUS REG
2141  003624  005202                    INC     R2              ;NEXT ADDRESS
2142  003626  022702  000010            CMP     #10,R2          ;ALL DONE?
```

```
2143  003632  001362                        BNE     1$              ;BR IF NO
2144  003634  012602                        MOV     (SP)+,R2        ;RESTORE R2
2145  003636  000205                        RTS     R5              ;RETURN
2146
2147
2148  003640                          MEMLD:
2149                                        ;THIS SUBROUTINE LOADS THE FIRST 8 LOCATIONS OF MAIN
2150                                        ;MEMORY WITH THIS DATA: 0,-1,,0,-1,125,252,125,252
2151
2152  003640  013637  002550                MOV     @(SP)+,$TMP0    ;PUT POINTER TO DATA IN R0
2153  003644  062746  000002                ADD     #2,-(SP)        ;ADJUST STACK
2154
2155  003650  013700  002550          MEMLD2: MOV     $TMP0,R0        ;GET ADDR.
2156  003654  012704  000010                MOV     #10,R4          ;DO 8 LOADS
2157  003660                                ROMCLK
2158  003660  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
2159  003664  010000                        010000                  ;MAR < 0
2160  003666                                ROMCLK                  ;CLR    MAR HI
2161  003666  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
2162  003672  004000                        004000
2163  003674  112077  177024          1$:   MOVB    (R0)+,@KMP04    ;LOAD PORT4
2164  003700                                ROMCLK
2165  003700  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
2166  003704  136500                        136500                  ;MOV DATA TO MEM, AUTO INC MAR
2167  003706  005304                        DEC     R4              ;DECREMENT COUNT
2168  003710  001371                        BNE     1$              ;BR IF NOT DONE
2169
2170  003712                                ROMCLK                  ;LOAD MEM ADDR. 0
2171  003712  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
2172  003716  010000                        10000
2173  003720  012703  000010                MOV     #10,R3          ;CHECK 8. MEM LOCS.
2174  003724  013700  002550                MOV     $TMP0,R0
2175  003730                          2$:   ROMCLK                  ;READ FROM MEM,PUT INTO PORT 4
2176  003730  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
2177  003734  055224                        55224
2178
2179  003736  112037  002636                MOVB    (R0)+,$GDDAT    ;EXPECTED.
2180  003742  117704  176756                MOVB    @KMP04,R4       ;RECIEVED.
2181  003746  123704  002636                CMPB    $GDDAT,R4       ;OK?
2182  003752  001414                        BEQ     3$
2183  003754                                ERROR   36
2184  003772  104455                        TRAP    C$ERDF
2185  003774  000044                        .WORD   36
2186  003776  005640                        .WORD   EM36
2187  004000  010432                        .WORD   ERR36
2188  004002  000402                        BR      4$
2189  004004  005303                  3$:   DEC     R3              ;CHECKED ALL?
2190  004006  001350                        BNE     2$              ;NO-DO NEXT ONE.
2191  004010                          4$:
2192  004010  000207                        RTS     PC              ;RETURN
2193
2194
2195  004012                          SPLD:
2196                                        ;THIS SUBROUTINE LOADS THE FIRST 8 SCRATCH PAD
2197                                        ;LOCATIONS WITH: 0,0,-1,-1,125,125,252,252
2198
```

```
2199  004012  013600                        MOV     @(SP)+,R0       ;PUT POINTER TO DATA IN R5
2200  004014  062746  000002                ADD     #2,-(SP)        ;ADJUST STACK
2201  004020  005004                        CLR     R4              ;START AT SP ADDRESS 0
2202  004022  112077  176676        1$:     MOVB    (R0)+,@KMP04    ;LOAD PORT4 WITH DATA
2203  004026  042737  000017  004044        BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
2204  004034  050437  004044                BIS     R4,2$           ;ADD ADDRESS TO INSTRUCTION
2205  004040                                ROMCLK
2206  004040  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
2207  004044  123100                2$:     123100                  ;MOVE DATA TO SP
2208  004046  005204                        INC     R4              ;INCREMENT COUNT
2209  004050  022704  000010                CMP     #10,R4          ;DONE YET?
2210  004054  001362                        BNE     1$              ;BR IF NO
2211  004056  000207                        RTS     PC              ;RETURN
2212
2213
2214  004060                        CLRC:
2215                                        ;THIS SUBROUTINE CLEARS THE MICRO PROCESSOR C BIT
2216
2217  004060                                ROMCLK
2218  004060  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
2219  004064  010000                        010000                  ;MAR_0
2220  004066                                ROMCLK
2221  004066  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
2222  004072  040400                        040400!<0*20>           ;CLEAR C BIT
2223  004074  000207                        RTS     PC              ;RETURN
2224
2225
2226  004076                        SETC:
2227                                        ;THIS SUBROUTINE SETS THE MICRO PROCESSOR C BIT
2228
2229  004076                                ROMCLK
2230  004076  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
2231  004102  010003                        010003                  ;MAR_3
2232  004104                                ROMCLK
2233  004104  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
2234  004110  040403                        040403!<0*20>           ;SET C BIT
2235  004112  000207                        RTS     PC              ;RETURN
2236
2237
2238
2239
```

```
2240                                    .SBTTL  GLOBAL ERROR REPORT SECTION
2241
2242                                    ;////////////////////////////////////////////////////////////////////
2243                                    ;/      THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
2244                                    ;/      THAT ARE USED IN MORE THAN ONE TEST.
2245                                    ;////////////////////////////////////////////////////////////////////
```

```
2246
2247
2248
2249  004114  047045  052045  047045  FM1:    .ASCIZ  /%N%T%N/
2250  004122  000
2251  004123  045  031517  051445  TFM1:   .ASCIZ  /%03%S5%03%S5%03%N2/
2252  004130  022465  031517  051445
2253  004136  022465  031517  047045
2254  004144  000062
2255  004146  047445  022466  031123  TFM2:   .ASCIZ  /%06%S2%06%N2/
2256  004154  047445  022466  031116
2257  004162  000
2258  004163  045  031517  051445  TFM5:   .ASCIZ  /%03%S5%03%N2/
2259  004170  022465  031517  047045
2260  004176  000062
2261  004200  047045  047445  022463  TFM27:  .ASCIZ  /%N%03%S5%06%S7%06%N2/
2262  004206  032523  047445  022466
2263  004214  033523  047445  022466
2264  004222  031116  000
2265  004225  045  022516  043101  TFM37:  .ASCIZ  /%N%AFAILING ADDRESS IS: %06/
2266  004232  044501  044514  043516
2267  004240  040440  042104  042522
2268  004246  051523  044440  035123
2269  004254  022440  033117  000
2270
2271
```

```
 2272
 2273
 2274  004261     122  043505  051511  EM1:    .ASCIZ  &REGISTER ADDRESS TEST&
 2275  004266  042524  020122  042101
 2276  004274  051104  051505  020123
 2277  004302  042524  052123     000
 2278  004307     111  052502  025123  EM2:    .ASCIZ  &IBUS* REGISTER DUAL ADDRESSING TEST&
 2279  004314  051040  043505  051511
 2280  004322  042524  020122  052504
 2281  004330  046101  040440  042104
 2282  004336  042522  051523  047111
 2283  004344  020107  042524  052123
 2284  004352     000
 2285  004353     111  052502  020123  EM30:   .ASCIZ  ''IBUS REGISTER DUAL ADDRESSING TEST''
 2286  004360  042522  044507  052123
 2287  004366  051105  042040  040525
 2288  004374  020114  042101  051104
 2289  004402  051505  044523  043516
 2290  004410  052040  051505  000124
 2291  004416  051102  051040  043505  EM3:    .ASCIZ  /BR REGISTER DATA TEST/
 2292  004424  051511  042524  020122
 2293  004432  040504  040524  052040
 2294  004440  051505  000124
```

```
  2295  004444  041523  04C522  041524  EM4:    .ASCIZ  /SCRATCH PAD DATA TEST/
  2296  004452  020110  040520  020104
  2297  004460  040504  040524  052040
  2298  004466  051505  000124
```

```
  2299  004472  041523  04C522  041524  EM5:    .ASCIZ  /SCRATCH PAD DUAL ADDRESSING TEST/
  2300  004500  020110  040520  020104
  2301  004506  052504  046101  040440
  2302  004514  042104  042522  051523
  2303  004522  047111  020107  042524
  2304  004530  052123     000
```

```
 2305  004533     115  044501  020116  EM6:    .ASCIZ  /MAIN MEMORY DATA TEST/
 2306  004540  042515  047515  054522
 2307  004546  042040  052101  020101
 2308  004554  042524  052123     000
 2309  004561     115  044501  020116  EM7:    .ASCIZ  /MAIN MEMORY DUAL ADDRESSING TEST/
 2310  004566  042515  047515  054522
 2311  004574  042040  040525  020114
 2312  004602  042101  051104  051505
 2313  004610  044523  043516  052040
 2314  004616  051505  000124
```

```
 2315  004622  052501  047524  046440  EM10:   .ASCIZ  /AUTO MARINC FUNCTION TEST/
 2316  004630  051101  047111  020103
 2317  004636  052506  041516  044524
 2318  004644  047117  052040  051505
 2319  004652  000124
 2320  004654  050116  020122  042524  EM11:   .ASCIZ  /NPR TEST/
 2321  004662  052123     000
 2322  004665     115  046125  044524  EM12:   .ASCIZ  /MULTIPLE NPR TEST/
 2323  004672  046120  020105  050116
 2324  004700  020122  042524  052123
 2325  004706     000
 2326  004707     116  047117  042440  EM13:   .ASCIZ  /NON EX MEM FAILED/
 2327  004714  020130  042515  020115
 2328  004722  040506  046111  042105
 2329  004730     000
 2330  004731     120  047522  051107  EM14:   .ASCIZ  /PROGRAM CLOCK TEST/
 2331  004736  046501  041440  047514
 2332  004744  045503  052040  051505
 2333  004752  000124
 2334  004754  046101  020125  052506  EM15:   .ASCIZ  /ALU FUNCTION WITH C BIT CLEAR TEST/
 2335  004762  041516  044524  047117
 2336  004770  053440  052111  020110
 2337  004776  020103  044502  020124
 2338  005004  046103  040505  020122
 2339  005012  042524  052123     000
```

```
2340  005017    120  053517  051105  EM16:   .ASCIZ  /POWER FAIL: BUS INIT WAS NOT BLOCKED/
2341  005024 043040  044501  035114
2342  005032 041040  051525  044440
2343  005040 044516  020124  040527
2344  005046 020123  047516  020124
2345  005054 046102  041517  042513
2346  005062 000104
2347  005064                        EM35:
2348  005064 047506  041522  020105  EM17:   .ASCIZ  /FORCE POWER FAIL ERROR/
2349  005072 047520  042527  020122
2350  005100 040506  046111  042440
2351  005106 051122  051117     000
2352  005113    116  044517  020523  EM20:   .ASCIZ  /NOISE TEST ON IBUS*,IBUS,SPAD,MEMORY/
2353  005120 052040  051505  020124
2354  005126 047117  044440  052502
2355  005134 025123  044454  052502
2356  005142 026123  050123  042101
2357  005150 046454  046505  051117
2358  005156 000131
2359  005160 046101  020125  020103  EM21:   .ASCIZ  /ALU C BIT TEST FAILURE/
2360  005166 044502  020124  042524
2361  005174 052123  043040  044501
2362  005202 052514  042522     000
2363  005207    124  046511  020105  EM22:   .ASCIZ  /TIME OUT ERROR/
2364  005214 052517  020124  051105
2365  005222 047522  000122
2366  005226 046101  020125  052506  EM23:   .ASCIZ  /ALU FUNCTION TEST WITH C BIT SET/
2367  005234 041516  044524  047117
2368  005242 052040  051505  020124
2369  005250 044527  044124  041440
2370  005256 041040  052111  051440
2371  005264 052105     000
2372  005267    125  041520  051440  EM24:   .ASCIZ  /UPC SEQUENCE ERROR/
2373  005274 050505  042525  041516
2374  005302 020105  051105  047522
2375  005310 000122
2376  005312 050125  043040  044501  EM31:   .ASCIZ  'UP FAILED TO INTERRUPT''
2377  005320 042514  020104  047524
2378  005326 044440  052116  051105
2379  005334 052522  052120     000
2380  005341    125  020120  047111  EM32:   .ASCIZ  'UP INTERRUPTED TO WRONG VECTOR''
2381  005346 042524  051122  050125
2382  005354 042524  020104  047524
2383  005362 053440  047522  043516
2384  005370 053040  041505  047524
2385  005376 000122
2386  005400 047125  054105  042520  EM33:   .ASCIZ  'UNEXPECTED INTERRUPT FROM UP''
2387  005406 052103  042105  044440
2388  005414 052116  051105  052522
2389  005422 052120  043040  047522
2390  005430 020115  050125     000
2391  005435    101  052514  043040  EM34:   .ASCIZ  ''ALU FLAG TEST''
2392  005442 040514  020107  042524
2393  005450 052123     000
2394  005453    110  046105  020114  EM25:   .ASCIZ  /HELL RAISER TEST/
2395  005460 040522  051511  051105
```

```
2396  005466  052040  051505  000124
2397  005474  040515  047111  040524  EM26:   .ASCIZ  /MAINTANCE REGISTER ERROR/
2398  005502  041516  020105  042522
2399  005510  044507  052123  051105
2400  005516  042440  051122  051117
2401  005524     000
2402  005525     111  052502  025123  EM27:   .ASCIZ  'IBUS* WRITE/READ ERROR'
2403  005532  053440  044522  042524
2404  005540  051057  040505  020104
2405  005546  051105  047522  000122
2406  005554  047111  052123  052522  EM28:   .ASCIZ  /INSTRUCTION TEST FAILURE/
2407  005562  052103  047511  020116
2408  005570  042524  052123  043040
2409  005576  044501  052514  042522
2410  005604     000
2411  005605     111  052502  027523  EM29:   .ASCIZ  'IBUS/OBUS WRITE/READ ERROR'
2412  005612  041117  051525  053440
2413  005620  044522  042524  051057
2414  005626  040505  020104  051105
2415  005634  047522  000122
2416
2417  005640  047511  020120  040515  EM36:   .ASCIZ  'IOP MAIN MEM. LOAD ERROR-RUN MCPU MEM. DIAG.'
2418  005646  047111  046440  046505
2419  005654  020056  047514  042101
2420  005662  042440  051122  051117
2421  005670  051055  047125  046440
2422  005676  050103  020125  042515
2423  005704  027115  042040  040511
2424  005712  027107     000
2425  005715     000                  EM37:   .ASCIZ  //
2426
2427
```

2428

```
  2429  005716     000                 DHO:     .ASCIZ  //
```

```
2430  005717     107  047517  020104  DH1:    .ASCIZ  /GOOD    BAD     REGISTER/
2431  005724  020040  041040  042101
2432  005732  020040  020040  051040
2433  005740  043505  051511  042524
2434  005746  000122
2435  005750  047507  042117  020040  DH2:    .ASCIZ  /GOOD    BAD/
2436  005756  020040  040502  000104
2437  005764  047507  042117  020040  DH3:    .ASCIZ  /GOOD    BAD     ADDRESS/
2438  005772  020040  040502  020104
2439  006000  020040  020040  042101
2440  006006  051104  051505  000123
2441  006014  047507  042117  020040  DH4:    .ASCIZ  /GOOD    BAD/
2442  006022  020040  040502  000104
2443  006030  042522  027107  042440  DH27:   .ASCIZ  /REG. EXPECTED FOUND/
2444  006036  050130  041505  042524
2445  006044  020104  047506  047125
2446  006052  000104
2447
2448
2449
2450
2451                                  .EVEN
```

```
2452
2453
2454
```

```
  2455                                  ;-----------------------------------------------------------------------------
  2456                                  ; MACRO'S NEEDED TO REPORT ERRORS
```

```
2457                                    ;-------------------------------------------------------------------------------
2458                                    .MACRO  MDT0
2459                                    .ENDM
2460
2461                                    .MACRO  MDT1
2462                                            PRINTB #TFM1,$GDDAT,$BDDAT,$GDADR
2463                                    .ENDM
2464
2465                                    .MACRO MDT2
```

```
2466                                    PRINTB #TFM2,$GDDAT,$BDDAT
2467                            .ENDM
2468
2469                            .MACRO  MDT5
2470                                    PRINTB  #TFM5,$GDDAT,$BDDAT
2471                            .ENDM
```

```
2472                                    .MACRO  MDT27
2473                                            PRINTB #TFM27,MR0,$GDDAT,$BDDAT
2474                                    .ENDM
2475
2476                                    .MACRO  $MD,ERNB,ERHM,ERFM
2477                                    .NLIST
2478                                    ; ERNB = ERROR NUMBER
2479                                    ; ERFM = FORMAT NUMBER
2480                                    ; ERHM = HEADER NUMBER
2481                                    .LIST
2482                                            BGNMSG  ERR'ERNB'
2483                                            PRINTB  #FM1,#DH'ERHM'
2484                                            MDT'ERFM'
2485                                            ENDMSG
```

```
2486                                    .ENDM
2487
2488
2439
2490
2491  006054                                    $MD     1,2,2
2492  006054                            ERR1::
2493  006054  012746  005750                    MOV     #DH2,-(SP)
2494  006060  012746  004114                    MOV     #FM1,-(SP)
2495  006064  012746  000002                    MOV     #2,-(SP)
2496  006070  010600                            MOV     SP,R0
2497  006072  104414                            TRAP    C$PNTB
2498  006074  062706  000006                    ADD     #6,SP
2499  006100  013746  002640                    MOV     $BDDAT,-(SP)
2500  006104  013746  002636                    MOV     $GDDAT,-(SP)
2501  006110  012746    4146                    MOV     #TFM2,-(SP)
2502  006114  012746  000003                    MOV     #3,-(SP)
2503  006120  010600                            MOV     SP,R0
2504  006122  104414                            TRAP    C$PNTB
2505  006124  062706  000010                    ADD     #10,SP
2506  006130                            L10003:
2507  006130  104423                            TRAP    C$MSG
2508  006132                                    $MD     2,2,2
2509  006132                            ERR2::
2510  006132  012746  005750                    MOV     #DH2,-(SP)
2511  006136  012746  004114                    MOV     #FM1,-(SP)
2512  006142  012746  000002                    MOV     #2,-(SP)
2513  006146  010600                            MOV     SP,R0
2514  006150  104414                            TRAP    C$PNTB
2515  006152  062706  000006                    ADD     #6,SP
2516  006156  013746  002640                    MOV     $BDDAT,-(SP)
2517  006162  013746  002636                    MOV     $GDDAT,-(SP)
2518  006166  012746  004146                    MOV     #TFM2,-(SP)
2519  006172  012746  000003                    MOV     #3,-(SP)
2520  006176  010600                            MOV     SP,R0
2521  006200  104414                            TRAP    C$PNTB
2522  006202  062706  000010                    ADD     #10,SP
2523  006206                            L10004:
2524  006206  104423                            TRAP    C$MSG
2525  006210                                    $MD     3,2,2
2526  006210                            ERR3::
2527  006210  012746  005750                    MOV     #DH2,-(SP)
2528  006214  012746  004114                    MOV     #FM1,-(SP)
2529  006220  012746  000002                    MOV     #2,-(SP)
2530  006224  010600                            MOV     SP,R0
2531  006226  104414                            TRAP    C$PNTB
2532  006230  062706  000006                    ADD     #6,SP
2533  006234  013746  002640                    MOV     $BDDAT -(SF)
2534  006240  013746  002636                    MOV     $GDDAT,-(SP)
2535  006244  012746  004146                    MOV     #TFM2,-(SP)
2536  006250  012746  000003                    MOV     #3,-(SP)
2537  006254  010600                            MOV     SP,R0
2538  006256  104414                            TRAP    C$PNTB
2539  006260  062706  000010                    ADD     #10,SP
2540  006264                            L10005:
2541  006264  104423                            TRAP    C$MSG
```

```
2542  006266                                    $MD     4,1,1
2543  006266                              ERR4::
2544  006266  012746  005717                    MOV     #DH1,-(SP)
2545  006272  012746  004114                    MOV     #FM1,-(SP)
2546  006276  012746  000002                    MOV     #2,-(SP)
2547  006302  010600                            MOV     SP,R0
2548  006304  104414                            TRAP    C$PNTB
2549  006306  062706  000006                    ADD     #6,SP
2550  006312  013746  002632                    MOV     $GDADR,-(SP)
2551  006316  013746  002640                    MOV     $BDDAT,-(SP)
2552  006322  013746  002636                    MOV     $GDDAT,-(SP)
2553  006326  012746  004123                    MOV     #TFM1,-(SP)
2554  006332  012746  000004                    MOV     #4,-(SP)
2555  006336  010600                            MOV     SP,R0
2556  006340  104414                            TRAP    C$PNTB
2557  006342  062706  000012                    ADD     #12,SP
2558  006346                              L10006:
2559  006346  104423                            TRAP    C$MSG
2560  006350                                    $MD     5,1,1
2561  006350                              ERR5::
2562  006350  012746  005717                    MOV     #DH1,-(SP)
2563  006354  012746  004114                    MOV     #FM1,-(SP)
2564  006360  012746  000002                    MOV     #2,-(SP)
2565  006364  010600                            MOV     SP,R0
2566  006366  104414                            TRAP    C$PNTB
2567  006370  062706  000006                    ADD     #6,SP
2568  006374  013746  002632                    MOV     $GDADR,-(SP)
2569  006400  013746  002640                    MOV     $BDDAT,-(SP)
2570  006404  013746  002636                    MOV     $GDDAT,-(SP)
2571  006410  012746  004123                    MOV     #TFM1,-(SP)
2572  006414  012746  000004                    MOV     #4,-(SP)
2573  006420  010600                            MOV     SP,R0
2574  006422  104414                            TRAP    C$PNTB
2575  006424  062706  000012                    ADD     #12,SP
2576  006430                              L10007:
2577  006430  104423                            TRAP    C$MSG
2578  006432                                    $MD     6,3,1
2579  006432                              ERR6::
2580  006432  012746  005764                    MOV     #DH3,-(SP)
2581  006436  012746  004114                    MOV     #FM1,-(SP)
2582  006442  012746  000002                    MOV     #2,-(SP)
2583  006446  010600                            MOV     SP,R0
2584  006450  104414                            TRAP    C$PNTB
2585  006452  062706  000006                    ADD     #6,SP
2586  006456  013746  002632                    MOV     $GDADR,-(SP)
2587  006462  013746  002640                    MOV     $BDDAT,-(SP)
2588  006466  013746  002636                    MOV     $GDDAT,-(SP)
2589  006472  012746  004123                    MOV     #TFM1,-(SP)
2590  006476  012746  000004                    MOV     #4,-(SP)
2591  006502  010600                            MOV     SP,R0
2592  006504  104414                            TRAP    C$PNTB
2593  006506  062706  000012                    ADD     #12,SP
2594  006512                              L10010:
2595  006512  104423                            TRAP    C$MSG
2596  006514                                    $MD     7,3,1
2597  006514                              ERR7::
```

```
2598  006514  012746  005764                  MOV     #DH3,-(SP)
2599  006520  012746  004114                  MOV     #FM1,-(SP)
2600  006524  012746  000002                  MOV     #2,-(SP)
2601  006530  010600                          MOV     SP,R0
2602  006532  104414                          TRAP    C$PNTB
2603  006534  062706  000006                  ADD     #6,SP
2604  006540  013746  002632                  MOV     $GDADR,-(SP)
2605  006544  013746  002640                  MOV     $BDDAT,-(SP)
2606  006550  013746  002636                  MOV     $GDDAT,-(SP)
2607  006554  012746  004123                  MOV     #TFM1,-(SP)
2608  006560  012746  000004                  MOV     #4,-(SP)
2609  006564  010600                          MOV     SP,R0
2610  006566  104414                          TRAP    C$PNTB
2611  006570  062706  000012                  ADD     #12,SP
2612  006574                          L10011:
2613  006574  104423                          TRAP    C$MSG
2614  006576                                  $MD     10,3,1
2615  006576                          ERR10::
2616  006576  012746  005764                  MOV     #DH3,-(SP)
2617  006602  012746  004114                  MOV     #FM1,-(SP)
2618  006606  012746  000002                  MOV     #2,-(SP)
2619  006612  010600                          MOV     SP,R0
2620  006614  104414                          TRAP    C$PNTB
2621  006616  062706  000006                  ADD     #6,SP
2622  005622  013746  002632                  MOV     $GDADR,-(SP)
2623  006626  013746  002640                  MOV     $BDDAT,-(SP)
2624  006632  013746  002636                  MOV     $GDDAT,-(SP)
2625  006636  012746  004123                  MOV     #TFM1,-(SP)
2626  006642  012746  000004                  MOV     #4,-(SP)
2627  006646  010600                          MOV     SP,R0
2628  006650  104414                          TRAP    C$PNTB
2629  006652  062706  000012                  ADD     #12,SP
2630  006656                          L10012:
2631  006656  104423                          TRAP    C$MSG
2632  006660                                  $MD     11,2,2
2633  006660                          ERR11::
2634  006660  012746  005750                  MOV     #DH2,-(SP)
2635  006664  012746  004114                  MOV     #FM1,-(SP)
2636  006670  012746  000002                  MOV     #2,-(SP)
2637  006674  010600                          MOV     SP,R0
2638  006676  104414                          TRAP    C$PNTB
2639  006700  062706  000006                  ADD     #6,SP
2640  006704  013746  002640                  MOV     $BDDAT,-(SP)
2641  006710  013746  002636                  MOV     $GDDAT,-(SP)
2642  006714  012746  004146                  MOV     #TFM2,-(SP)
2643  006720  012746  000003                  MOV     #3,-(SP)
2644  006724  010600                          MOV     SP,R0
2645  006726  104414                          TRAP    C$PNTB
2646  006730  062706  000010                  ADD     #10,SP
2647  006734                          L10013:
2648  006734  104423                          TRAP    C$MSG
2649  006736                                  $MD     12,2,2
2650  006736                          ERR12::
2651  006736  012746  005750                  MOV     #DH2,-(SP)
2652  006742  012746  004114                  MOV     #FM1,-(SP)
2653  006746  012746  000002                  MOV     #2,-(SP)
```

```
 2654  006752  010600                          MOV     SP,R0
 2655  006754  104414                          TRAP    C$PNTB
 2656  006756  062706  000006                  ADD     #6,SP
 2657  006762  013746  002640                  MOV     $BDDAT,-(SP)
 2658  006766  013746  002636                  MOV     $GDDAT,-(SP)
 2659  006772  012746  004146                  MOV     #TFM2,-(SP)
 2660  006776  012746  000003                  MOV     #3,-(SP)
 2661  007002  010600                          MOV     SP,R0
 2662  007004  104414                          TRAP    C$PNTB
 2663  007006  062706  000010                  ADD     #10,SP
 2664  007012                          L10014:
 2665  007012  104423                          TRAP    C$MSG
 2666  007014                                  $MD     13,0,0
 2667  007014                          ERR13::
 2668  007014  012746  005716                  MOV     #DH0,-(SP)
 2669  007020  012746  004114                  MOV     #FM1,-(SP)
 2670  007024  012746  000002                  MOV     #2,-(SP)
 2671  007030  010600                          MOV     SP,R0
 2672  007032  104414                          TRAP    C$PNTB
 2673  007034  062706  000006                  ADD     #6,SP
 2674  007040                          L10015:
 2675  007040  104423                          TRAP    C$MSG
```

```
 2676  007042                                    $MD     14,2,2
 2677  007042                              ERR14::
 2678  007042  012746  005750                    MOV     #DH2,-(SP)
 2679  007046  012746  004114                    MOV     #FM1,-(SP)
 2680  007052  012746  000002                    MOV     #2,-(SP)
 2681  007056  010600                            MOV     SP,R0
 2682  007060  104414                            TRAP    C$PNTB
 2683  007062  062706  000006                    ADD     #6,SP
 2684  007066  013746  002640                    MOV     $BDDAT,-(SP)
 2685  007072  013746  002636                    MOV     $GDDAT,-(SP)
 2686  007076  012746  004146                    MOV     #TFM2,-(SP)
 2687  007102  012746  000003                    MOV     #3,-(SP)
 2688  007106  010600                            MOV     SP,R0
 2689  007110  104414                            TRAP    C$PNTB
 2690  007112  062706  000010                    ADD     #10,SP
 2691  007116                              L10016:
 2692  007116  104423                            TRAP    C$MSG
```

```
2693  007120                                        $MD     15,4,5
2694  007120                                ERR15::
2695  007120  012746  006014                        MOV     #DH4,-(SP)
2696  007124  012746  004114                        MOV     #FM1,-(SP)
2697  007130  012746  000002                        MOV     #2,-(SP)
2698  007134  010600                                MOV     SP,R0
2699  007136  104414                                TRAP    C$PNTB
2700  007140  062706  000006                        ADD     #6,SP
2701  007144  013746  002640                        MOV     $BDDAT,-(SP)
2702  007150  013746  002636                        MOV     $GDDAT,-(SP)
2703  007154  012746  004163                        MOV     #TFM5,-(SP)
2704  007160  012746  000003                        MOV     #3,-(SP)
2705  007164  010600                                MOV     SP,R0
2706  007166  104414                                TRAP    C$PNTB
2707  007170  062706  000010                        ADD     #10,SP
2708  007174                                L10017:
2709  007174  104423                                TRAP    C$MSG
2710  007176                                        $MD     16,0,0
2711  007176                                ERR16::
2712  007176  012746  005716                        MOV     #DH0,-(SP)
2713  007202  012746  004114                        MOV     #FM1,-(SP)
2714  007206  012746  000002                        MOV     #2,-(SP)
2715  007212  010600                                MOV     SP,R0
2716  007214  104414                                TRAP    C$PNTB
2717  007216  062706  000006                        ADD     #6,SP
2718  007222                                L10020:
2719  007222  104423                                TRAP    C$MSG
2720  007224                                        $MD     17,0,0
2721  007224                                ERR17::
2722  007224  012746  005716                        MOV     #DH0,-(SP)
2723  007230  012746  004114                        MOV     #FM1,-(SP)
2724  007234  012746  000002                        MOV     #2,-(SP)
2725  007240  010600                                MOV     SP,R0
2726  007242  104414                                TRAP    C$PNTB
2727  007244  062706  000006                        ADD     #6,SP
2728  007250                                L10021:
2729  007250  104423                                TRAP    C$MSG
2730  007252                                        $MD     20,2,2
2731  007252                                ERR20::
2732  007252  012746  005750                        MOV     #DH2,-(SP)
2733  007256  012746  004114                        MOV     #FM1,-(SP)
2734  007262  012746  000002                        MOV     #2,-(SP)
2735  007266  010600                                MOV     SP,R0
2736  007270  104414                                TRAP    C$PNTB
2737  007272  062706  000006                        ADD     #6,SP
2738  007276  013746  002640                        MOV     $BDDAT,-(SP)
2739  007302  013746  002636                        MOV     $GDDAT,-(SP)
2740  007306  012746  004146                        MOV     #TFM2,-(SP)
2741  007312  012746  000003                        MOV     #3,-(SP)
2742  007316  010600                                MOV     SP,R0
2743  007320  104414                                TRAP    C$PNTB
2744  007322  062706  000010                        ADD     #10,SP
2745  007326                                L10022:
2746  007326  104423                                TRAP    C$MSG
2747  007330                                        $MD     21,0,0
2748  007330                                ERR21::
```

```
 2749  007330  012746  005716                MOV     #DH0,-(SP)
 2750  007334  012746  004114                MOV     #FM1,-(SP)
 2751  007340  012746  000002                MOV     #2,-(SP)
 2752  007344  010600                        MOV     SP,R0
 2753  007346  104414                        TRAP    C$PNTB
 2754  007350  062706  000006                ADD     #6,SP
 2755  007354                        L10023:
 2756  007354  104423                        TRAP    C$MSG
 2757  007356                                $MD     22,0,0
 2758  007356                        ERR22::
 2759  007356  012746  005716                MOV     #DH0,-(SP)
 2760  007362  012746  004114                MOV     #FM1,-(SP)
 2761  007366  012746  000002                MOV     #2,-(SP)
 2762  007372  010600                        MOV     SP,R0
 2763  007374  104414                        TRAP    C$PNTB
 2764  007376  062706  000006                ADD     #6,SP
 2765  007402                        L10024:
 2766  007402  104423                        TRAP    C$MSG
 2767  007404                                $MD     23,4,5
 2768  007404                        ERR23::
 2769  007404  012746  006014                MOV     #DH4,-(SP)
 2770  007410  012746  004114                MOV     #FM1,-(SP)
 2771  007414  012746  000002                MOV     #2,-(SP)
 2772  007420  010600                        MOV     SP,R0
 2773  007422  104414                        TRAP    C$PNTB
 2774  007424  062706  000006                ADD     #6,SP
 2775  007430  013746  002640                MOV     $BDDAT,-(SP)
 2776  007434  013746  002636                MOV     $GDDAT,-(SP)
 2777  007440  012746  004163                MOV     #TFM5,-(SP)
 2778  007444  012746  000003                MOV     #3,-(SP)
 2779  007450  010600                        MOV     SP,R0
 2780  007452  104414                        TRAP    C$PNTB
 2781  007454  062706  000010                ADD     #10,SP
 2782  007460                        L10025:
 2783  007460  104423                        TRAP    C$MSG
 2784  007462                                $MD     24,0,0
 2785  007462                        ERP24::
 2786  007462  012746  005716                MOV     #DH0,-(SP)
 2787  007466  012746  004114                MOV     #FM1,-(SP)
 2788  007472  012746  000002                MOV     #2,-(SP)
 2789  007476  010600                        MOV     SP,R0
 2790  007500  104414                        TRAP    C$PNTB
 2791  007502  062706  000006                ADD     #6,SP
 2792  007506                        L10026:
 2793  007506  104423                        TRAP    C$MSG
 2794  007510                                $MD     25,2,2
 2795  007510                        ERR25::
 2796  007510  012746  005750                MOV     #DH2,-(SP)
 2797  007514  012746  004114                MOV     #FM1,-(SP)
 2798  007520  012746  000002                MOV     #2,-(SP)
 2799  007524  010600                        MOV     SP,R0
 2800  007526  104414                        TRAP    C$PNTB
 2801  007530  062706  000006                ADD     #6,SP
 2802  007534  013746  002640                MOV     $BDDAT,-(SP)
 2803  007540  013746  002636                MOV     $GDDAT,-(SP)
 2804  007544  012746  004146                MOV     #TFM2,-(SP)
```

```
 2805  007550  012746  000003                 MOV     #3,-(SP)
 2806  007554  010600                         MOV     SP,R0
 2807  007556  104414                         TRAP    C$PNTB
 2808  007560  062706  000010                 ADD     #10,SP
 2809  007564                          L10027:
 2810  007564  104423                         TRAP    C$MSG
 2811  007566                                  $MD     26,2,2
 2812  007566                          ERR26::
 2813  007566  012746  005750                 MOV     #DH2,-(SP)
 2814  007572  012746  004114                 MOV     #FM1,-(SP)
 2815  007576  012746  000002                 MOV     #2,-(SP)
 2816  007602  010600                         MOV     SP,R0
 2817  007604  104414                         TRAP    C$PNTB
 2818  007606  062706  000006                 ADD     #6,SP
 2819  007612  013746  002640                 MOV     $BDDAT,-(SP)
 2820  007616  013746  002636                 MOV     $GDDAT,-(SP)
 2821  007622  012746  004146                 MOV     #TFM2,-(SP)
 2822  007626  012746  000003                 MOV     #3,-(SP)
 2823  007632  010600                         MOV     SP,R0
 2824  007634  104414                         TRAP    C$PNTB
 2825  007636  062706  000010                 ADD     #10,SP
 2826  007642                          L10030:
 2827  007642  104423                         TRAP    C$MSG
 2828  007644                                  $MD     27,27,27
 2829  007644                          ERR27::
 2830  007644  012746  006030                 MOV     #DH27,-(SP)
 2831  007650  012746  004114                 MOV     #FM1,-(SP)
 2832  007654  012746  000002                 MOV     #2,-(SP)
 2833  007660  010600                         MOV     SP,R0
 2834  007662  104414                         TRAP    C$PNTB
 2835  007664  062706  000006                 ADD     #6,SP
 2836  007670  013746  002640                 MOV     $BDDAT,-(SP)
 2837  007674  013746  002636                 MOV     $GDDAT,-(SP)
 2838  007700  013746  002624                 MOV     MR0,-(SP)
 2839  007704  012746  004200                 MOV     #TFM27,-(SP)
 2840  007710  012746  000004                 MOV     #4,-(SP)
 2841  007714  010600                         MOV     SP,R0
 2842  007716  104414                         TRAP    C$PNTB
 2843  007720  062706  000012                 ADD     #12,SP
 2844  007724                          L10031:
 2845  007724  104423                         TRAP    C$MSG
 2846  007726                                  $MD     28,2,2
 2847  007726                          ERR28::
 2848  007726  012746  005750                 MOV     #DH2,-(SP)
 2849  007732  012746  004114                 MOV     #FM1,-(SP)
 2850  007736  012746  000002                 MOV     #2,-(SP)
 2851  007742  010600                         MOV     SP,R0
 2852  007744  104414                         TRAP    C$PNTB
 2853  007746  062706  000006                 ADD     #6,SP
 2854  007752  013746  002640                 MOV     $BDDAT,-(SP)
 2855  007756  013746  002636                 MOV     $GDDAT,-(SP)
 2856  007762  012746  004146                 MOV     #TFM2,-(SP)
 2857  007766  012746  000003                 MOV     #3,-(SP)
 2858  007772  010600                         MOV     SP,R0
 2859  007774  104414                         TRAP    C$PNTB
 2860  007776  062706  000010                 ADD     #10,SP
```

```
2861  010002                                L10032:
2862  010002  104423                              TRAP    C$MSG
2863  010004                                       $MD     29,27,27
2864  010004                                ERR29::
2865  010004  012746  006030                       MOV     #DH27,-(SP)
2866  010010  012746  004114                       MOV     #FM1,-(SP)
2867  010014  012746  000002                       MOV     #2,-(SP)
2868  010020  010600                               MOV     SP,R0
2869  010022  104414                               TRAP    C$PNTB
2870  010024  062706  000006                       ADD     #6,SP
2871  010030  013746  002640                       MOV     $BDDAT,-(SP)
2872  010034  013746  002636                       MOV     $GDDAT,-(SP)
2873  010040  013746  002624                       MOV     MR0,-(SP)
2874  010044  012746  004200                       MOV     #TFM27,-(SP)
2875  010050  012746  000004                       MOV     #4,-(SP)
2876  010054  010600                               MOV     SP,R0
2877  010056  104414                               TRAP    C$PNTB
2878  010060  062706  000012                       ADD     #12,SP
2879  010064                                L10033:
2880  010064  104423                              TRAP    C$MSG
2881  010066                                       $MD     30,2,2
2882  010066                                ERR30::
2883  010066  012746  005750                       MOV     #DH2,-(SP)
2884  010072  012746  004114                       MOV     #FM1,-(SP)
2885  010076  012746  000002                       MOV     #2,-(SP)
2886  010102  010600                               MOV     SP,R0
2887  010104  104414                               TRAP    C$PNTB
2888  010106  062706  000006                       ADD     #6,SP
2889  010112  013746  002640                       MOV     $BDDAT,-(SP)
2890  010116  013746  002636                       MOV     $GDDAT,-(SP)
2891  010122  012746  004146                       MOV     #TFM2,-(SP)
2892  010126  012746  000003                       MOV     #3,-(SP)
2893  010132  010600                               MOV     SP,R0
2894  010134  104414                               TRAP    C$PNTB
2895  010136  062706  000010                       ADD     #10,SP
2896  010142                                L10034:
2897  010142  104423                              TRAP    C$MSG
2898  010144                                       $MD     31,0,0
2899  010144                                ERR31::
2900  010144  012746  005716                       MOV     #DH0,-(SP)
2901  010150  012746  004114                       MOV     #FM1,-(SP)
2902  010154  012746  000002                       MOV     #2,-(SP)
2903  010160  010600                               MOV     SP,R0
2904  010162  104414                               TRAP    C$PNTB
2905  010164  062706  000006                       ADD     #6,SP
2906  010170                                L10035:
2907  010170  104423                              TRAP    C$MSG
2908  010172                                       $MD     32,0,0
2909  010172                                ERR32::
2910  010172  012746  005716                       MOV     #DH0,-(SP)
2911  010176  012746  004114                       MOV     #FM1,-(SP)
2912  010202  012746  000002                       MOV     #2,-(SP)
2913  010206  010600                               MOV     SP,R0
2914  010210  104414                               TRAP    C$PNTB
2915  010212  062706  000006                       ADD     #6,SP
2916  010216                                L10036:
```

```
 2917  010216  104423                          TRAP    C$MSG
 2918  010220                                  $MD     33,2,2
 2919  010220                          ERR33::
 2920  010220  012746  005750                  MOV     #DH2,-(SP)
 2921  010224  012746  004114                  MOV     #FM1,-(SP)
 2922  010230  012746  000002                  MOV     #2,-(SP)
 2923  010234  010600                          MOV     SP,R0
 2924  010236  104414                          TRAP    C$PNTB
 2925  010240  062706  000006                  ADD     #6,SP
 2926  010244  013746  002640                  MOV     $BDDAT,-(SP)
 2927  010250  013746  002636                  MOV     $GDDAT,-(SP)
 2928  010254  012746  004146                  MOV     #TFM2,-(SP)
 2929  010260  012746  000003                  MOV     #3,-(SP)
 2930  010264  010600                          MOV     SP,R0
 2931  010266  104414                          TRAP    C$PNTB
 2932  010270  062706  000010                  ADD     #10,SP
 2933  010274                          L10037:
 2934  010274  104423                          TRAP    C$MSG
 2935  010276                                  $MD     34,2,2
 2936  010276                          ERR34::
 2937  010276  012746  005750                  MOV     #DH2,-(SP)
 2938  010302  012746  004114                  MOV     #FM1,-(SP)
 2939  010306  012746  000002                  MOV     #2,-(SP)
 2940  010312  010600                          MOV     SP,R0
 2941  010314  104414                          TRAP    C$PNTB
 2942  010316  062706  000006                  ADD     #6,SP
 2943  010322  013746  002640                  MOV     $BDDAT,-(SP)
 2944  010326  013746  002636                  MOV     $GDDAT,-(SP)
 2945  010332  012746  004146                  MOV     #TFM2,-(SP)
 2946  010336  012746  000003                  MOV     #3,-(SP)
 2947  010342  010600                          MOV     SP,R0
 2948  010344  104414                          TRAP    C$PNTB
 2949  010346  062706  000010                  ADD     #10,SP
 2950  010352                          L10040:
 2951  010352  104423                          TRAP    C$MSG
 2952  010354                                  $MD     35,2,2
 2953  010354                          ERR35::
 2954  010354  012746  005750                  MOV     #DH2,-(SP)
 2955  010360  012746  004114                  MOV     #FM1,-(SP)
 2956  010364  012746  000002                  MOV     #2,-(SP)
 2957  010370  010600                          MOV     SP,R0
 2958  010372  104414                          TRAP    C$PNTB
 2959  010374  062706  000006                  ADD     #6,SP
 2960  010400  013746  002640                  MOV     $BDDAT,-(SP)
 2961  010404  013746  002636                  MOV     $GDDAT,-(SP)
 2962  010410  012746  004146                  MOV     #TFM2,-(SP)
 2963  010414  012746  000003                  MOV     #3,-(SP)
 2964  010420  010600                          MOV     SP,R0
 2965  010422  104414                          TRAP    C$PNTB
 2966  010424  062706  000010                  ADD     #10,SP
 2967  010430                          L10041:
 2968  010430  104423                          TRAP    C$MSG
 2969  010432                                  $MD     36,2,2
 2970  010432                          ERR36::
 2971  010432  012746  005750                  MOV     #DH2,-(SP)
 2972  010436  012746  004114                  MOV     #FM1,-(SP)
```

```
 2973  010442  012746  000002                    MOV     #2,-(SP)
 2974  010446  010600                            MOV     SP,R0
 2975  010450  104414                            TRAP    C$PNTB
 2976  010452  062706  000006                    ADD     #6,SP
 2977  010456  013746  002640                    MOV     $BDDAT,-(SP)
 2978  010462  013746  002636                    MOV     $GDDAT,-(SP)
 2979  010466  012746  004146                    MOV     #TFM2,-(SP)
 2980  010472  012746  000003                    MOV     #3,-(SP)
 2981  010476  010600                            MOV     SP,R0
 2982  010500  104414                            TRAP    C$PNTB
 2983  010502  062706  000010                    ADD     #10,SP
 2984  010506                            L10042:
 2985  010506  104423                            TRAP    C$MSG
 2986
 2987  010510                                    BGNMSG  ERR37
 2988  010510                            ERR37::
 2989  010510                                    PRINTF  #FM1,#EM1
 2990  010510  012746  004261                    MOV     #EM1,-(SP)
 2991  010514  012746  004114                    MOV     #FM1,-(SP)
 2992  010520  012746  000002                    MOV     #2,-(SP)
 2993  010524  010600                            MOV     SP,R0
 2994  010526  104417                            TRAP    C$PNTF
 2995  010530  062706  000006                    ADD     #6,SP
 2996  010534                                    PRINTF  #TFM37,$GDADR
 2997  010534  013746  002632                    MOV     $GDADR,-(SP)
 2998  010540  012746  004225                    MOV     #TFM37,-(SP)
 2999  010544  012746  000002                    MOV     #2,-(SP)
 3000  010550  010600                            MOV     SP,R0
 3001  010552  104417                            TRAP    C$PNTF
 3002  010554  062706  000006                    ADD     #6,SP
 3003  010560                                    ENDMSG
 3004  010560                            L10043:
 3005  010560  104423                            TRAP    C$MSG
 3006
 3007
 3008
```

```
                        .SBTTL  REPORT CODING SECTION


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

010562                          BGNRPT
010562                  L$RPT::


010562                          EXIT    RPT
010562  000167                  .WORD   J$JMP
010564  000000                  .WORD   L10044-2-.


010566                          ENDRPT
010566                  L10044:
010566  104425                  TRAP    C$RPT
```

```
 3034                                   .SBTTL  INITIALIZE SECTION
 3035
 3036                                   ;////////////////////////////////////////////////////////////////////////
 3037                                   ;/ THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
 3038                                   ;/ AT THE BEGINNING OF EACH PASS.
 3039                                   ;////////////////////////////////////////////////////////////////////////
 3040
 3041  010570                                   BGNINIT
 3042  010570                           L$INIT::
 3043
 3044                                   ;INITIALIZE SUBROUTINE STACK
 3045  010570  012705  003130                   MOV     #SSTACK,R5
 3046                                   ;STORE BASE LEVEL PROGRAM STACK POINTER
 3047  010574  010637  002554                   MOV     SP,PSTACK
 3048  010600  005737  002646                   TST     FTIME
 3049  010604  001011                           BNE     1$
 3050  010606  013737  000004  002650           MOV     @#4,SAVE4
 3051  010614  013737  000006  002652           MOV     @#6,SAVE6
 3052  010622  012737  000001  002646           MOV     #1,FTIME
 3053  010630  013737  002650  000004  1$:      MOV     SAVE4,@#4
 3054  010636  013737  002652  000006           MOV     SAVE6,@#6
 3055                                   ;SEE IF PROGRAM JUST STARTED, BR IF YES
 3056  010644                                   READEF  #EF.START
 3057  010644  012700  000040                   MOV     #EF.START,R0
 3058  010650  104447                           TRAP    C$REFG
 3059  010652                                   BCOMPLETE       NEWST
 3060  010652  103414                           BCS     NEWST
 3061                                   ;SEE IF THIS IS A NEW PASS, BR IF YES
 3062  010654                                   READEF  #EF.NEW
 3063  010654  012700  000035                   MOV     #EF.NEW,R0
 3064  010660  104447                           TRAP    C$REFG
 3065  010662                                   BCOMPLETE       NEWST
 3066  010662  103410                           BCS     NEWST
 3067                                   ;SEE IF PROGRAM WAS JUST CONTINUED
 3068  010664                                   READEF  #EF.CONTINUE
 3069  010664  012700  000036                   MOV     #EF.CONTINUE,R0
 3070  010670  104447                           TRAP    C$REFG
 3071  010672                                   BCOMPLETE       ENDIT
 3072  010672  103570                           BCS     ENDIT
 3073
 3074                                   ;SEE IF PROGRAM JUST RESTARTED, BR IF NOT
 3075  010674                                   READEF  #EF.RESTART
 3076  010674  012700  000037                   MOV     #EF.RESTART,R0
 3077  010700  104447                           TRAP    C$REFG
 3078  010702                                   BNCOMPLETE      GETPRM
 3079  010702  103003                           BCC     GETPRM
 3080  010704                           NEWST:
 3081                                   ;RESET LOGICAL DEVICE TO -1
 3082  010704  012737  177777  002552           MOV     #-1,LOGDEV
 3083
 3084                                   ;GET UNIBUS ADRS, VECTOR, PRIORITY LEVEL, LINE UNIT,SWITCH
 3085                                   ;    PACKS, TEST CONNECTOR INFO. FOR THIS M8200,4,7 (CURRENT LOGICAL
 3086                                   ;    DEVICE).
 3087  010712                           GETPRM:
 3088  010712  005237  002552                   INC     LOGDEV
 3089  010716  023737  002552  002012           CMP     LOGDEV,L$UNIT
```

```
3090  010724  002367                          BGE     NEWST
3091  010726                                  GPHARD  LOGDEV,R1
3092  010726  013700  002552                  MOV     LOGDEV,R0
3093  010732  104442                          TRAP    C$GPHRD
3094  010734  010001                          MOV     R0,R1
3095  010736                                  BNCOMPLETE      GETPRM
3096  010736  103365                          BCC     GETPRM
3097                                  ;GET ADDRESS OF M8200,4,7
3098  010740  012137  002626                  MOV     (R1)+,WTYPE
3099  010744  011137  002716                  MOV     (R1),KMCSR
3100                                  ;GET POINTER TO M8200,4,7 CSR HI BYTE
3101  010750  011137  002720                  MOV     (R1),KMCSRH
3102  010754  005237  002720                  INC     KMCSRH
3103                                  ;GET POINTER TO M8200,4,7 CTL OUT REG
3104  010760  011137  002722                  MOV     (R1),KMCTL
3105  010764  062737  000002  002722          ADD     #2,KMCTL
3106                                  ;GET POINTER TO M8200,4,7 PORT REG - SEL 4
3107  010772  011137  002724                  MOV     (R1),KMPO4
3108  010776  062737  000004  002724          ADD     #4,KMPO4
3109                                  ;GET POINTER TO M8200,4,7 PORT REG - SEL 6
3110  011004  012137  002726                  MOV     (R1)+,KMPO6
3111  011010  062737  000006  002726          ADD     #6,KMPO6
3112                                  ;GET POINTER TO RCV VECTOR
3113  011016  011137  002706                  MOV     (R1),KMRVEC
3114                                  ;GET POINTER TO RCV PRIORITY LEVEL
3115  011022  011137  002710                  MOV     (R1),KMRLVL
3116  011026  062737  000002  002710          ADD     #2,KMRLVL
3117                                  ;GET POINTER TO TX VECTOR
3118  011034  011137  002712                  MOV     (R1),KMTVEC
3119  011040  062737  000004  002712          ADD     #4,KMTVEC
3120                                  ;GET POINTER TO TX PRIORITY LEVEL
3121  011046  011137  002714                  MOV     (R1),KMTLVL
3122  011052  062737  000006  002714          ADD     #6,KMTLVL
3123                                  ;PUT VECTOR INTO STAT1
3124  011060  012137  002700                  MOV     (R1)+,STAT1
3125                                  ;PUT PRIORITY INTO STAT1
3126  011064  052137  002700                  BIS     (R1)+,STAT1
3127                                  ;SEE IF NO LINE UNIT, SET BIT IF YES
3128  011070  005711                          TST     (R1)
3129  011072  001004                          BNE     50000$
3130  011074  052737  010000  002700          BIS     #BIT12,STAT1
3131  011102  000416                          BR      4$
3132  011104                                  50000$:
3133                                  ;SEE IF M8201 LINE UNIT, SET BIT IF YES
3134  011104  021127  000001                  CMP     (R1),#1
3135  011110  001001                          BNE     50001$
3136  011112  000412                          BR      4$
3137  011114                                  50001$:
3138                                  ;SEE IF M8202 LINE UNIT, SET BIT IF YES
3139  011114  021127  000002                  CMP     (R1),#2
3140  011120  001004                          BNE     50002$
3141  011122  052737  020000  002700          BIS     #BIT13,STAT1
3142  011130  000403                          BR      4$
3143  011132                                  50002$:
3144                                  ;SET BIT FOR M8203 LINE UNIT
3145  011132  052737  100000  002700          BIS     #BIT15,STAT1
```

```
3146  011140                              4$:
3147                                      ;SET BIT IN STAT1 FOR TEST CONNECTOR
3148  011140  056137  000006  002700            BIS     6(R1),STAT1
3149  011146  062701  000002                    ADD     #2,R1
3150                                      ;SET SWITCH PACK #1 IN STAT2 LOW BYTE
3151  011152  012137  002702                    MOV     (R1)+,STAT2
3152                                      ;SET SWITCH PACK #2 IN STAT2 HIGH BYTE
3153  011156  111137  002703                    MOVB    (R1),STAT2+1
3154
3155                                      ;INCREMENT LOGICAL JNIT (DEVICE) NUMBER
3156                                      ;       INC     LOGDEV
3157  011162  000240                            NOP
3158  011164  000240                            NOP
3159
3160  011166  012737  002000  002606            MOV     #2000,MEMSZ
3161  011174  005037  002630                    CLR     TYPE
3162  011200  123727  002626  000000            CMPB    WTYPE,#0
3163  011206  001422                            BEQ     ENDIT
3164  011210  123727  002626  000004            CMPB    WTYPE,#4        ;KMC?
3165  011216  001004                            BNE     5$
3166  011220  012737  000001  002630            MOV     #1,TYPE
3167  011226  000412                            BR      ENDIT
3168  011230  012737  003777  002606  5$:       MOV     #3777,MEMSZ
3169  011236  123727  002626  000006            CMPB    WTYPE,#6
3170  011244  001003                            BNE     ENDIT
3171  011246  012737  000001  002630            MOV     #1,TYPE
3172  011254                              ENDIT:
3173  011254                                    ENDINIT
3174  011254                              L10045:
3175  011254  104411                            TRAP    C$INIT
3176
3177                                      .EVEN
3178  011256                                    BGNAUTO
3179  011256                              L$AUTO::
3180                                            ;DEVICE DOES NOT HAVE A "READY"
3181  011256  013701  002716                    MOV     KMCSR,R1        ;R1 CONTAINS BASE M8200,4,7 ADDRESS
3182  011262  012705  000004                    MOV     #4,R5           ;4 REGISTERS TO BE TESTED
3183  011266  012737  011320  000004            MOV     #2$,4           ;SET UP TIMEOUT TRAP
3184  011274  012737  000340  000006            MOV     #340,6          ;LEVEL 7
3185  011302  005711                      1$:   TST     (R1)            ;REFERENCE DEVICE REGISTER
3186  011304  000240                            NOP
3187  011306  062701  000002                    ADD     #2,R1           ;NEXT REGISTER
3188  011312  005305                            DEC     R5              ;DEC REGISTER COUNT
3189  011314  001372                            BNE     1$              ;BR IF NOT LAST REGISTER
3190  011316  000407                            BR      3$
3191  011320  062706  000004            2$:     ADD     #4,SP
3192  011324  010137  002632                    MOV     R1,$GDADR
3193  011330                                    DODU    LOGDEV
3194  011330  013700  002552                    MOV     LOGDEV,R0
3195  011334  104451                            TRAP    C$DODU
3196
3197  011336  013737  002650  000004  3$:       MOV     SAVE4,4
3198  011344  013737  002652  000006            MOV     SAVE6,6
3199  011352                                    ENDAUTO
3200  011352                              L10046:
3201  011352  104461                            TRAP    C$AUTO
```

```
3202
```

```
                                .SBTTL  CLEANUP CODING SECTION

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

011354                                  BGNCLN
011354                          L$CLEAN::
011354                                  BRESET
011354  104433                          TRAP    C$RESET

011356                                  ENDCLN
011356                          L10047:
011356  104412                          TRAP    C$CLEAN
```

```
                                .SBTTL  DROP UNIT SECTION

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

011360                                  BGNDU
011360                          L$DU::
                                ;ISSUE UNIBUS RESET TO CLEAN UP
011360                                  BRESET
011360  104433                          TRAP    C$RESET
011362                                  ENDDU
011362                          L10050:
011362  104453                          TRAP    C$DU
```

```
                                       .SBTTL  ADD UNIT SECTION

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

011364                                         BGNAU
011364                                 L$AU::
011364                                         ENDAU
011364                                 L10051:
011364  104452                                 TRAP    C$AU
```

```
                                        .SBTTL  HARDWARE TESTS


011366                                          BADHEAD
                                                ;******************** TEST 1 ********************
                                                ;*VERIFY THAT REFERENCING UNIBUS DEVICE REGISTERS
                                                ;*DOES NOT CAUSE A TIME OUT TRAP
011366                                          BADHEAD
                                                ;******************** TEST 1 ********************

011366                                  BGNTST
011366                                  T1::
011366  013701  002716                          MOV     KMCSR,R1        ;R1 CONTAINS BASE M8200,4,7 ADDRESS
011372  012705  000004                          MOV     #4,R5           ;4 REGISTERS TO BE TESTED
011376  012737  011434  000004                  MOV     #2$,4           ;SET UP TIMEOUT TRAP
011404  012737  000340  000006                  MOV     #340,6          ;LEVEL 7
011412  005711                          1$:     TST     (R1)            ;REFERENCE DEVICE REGISTER
011414  000240                                  NOP
011416                                          ESCAPE TST
011416  104410                                  TRAP    C$ESCAPE
011420  000072                                  .WORD   L10052-.
011422  062701  000002                          ADD     #2,R1           ;NEXT REGISTER
011426  005305                                  DEC     R5              ;DEC REGISTER COUNT
011430  001370                                  BNE     1$              ;BR IF NOT LAST REGISTER
011432  000417                                  BR      3$
011434  062706  000004                  2$:     ADD     #4,SP
011440  010137  002632                          MOV     R1,$GDADR
011444                                          ERROR   37              ;TIME-OUT ERROR
011462  104455                                  TRAP    C$ERDF
011464  000045                                  .WORD   37
011466  005715                                  .WORD   EM37
011470  010510                                  .WORD   ERR37

011472  013737  002650  000004          3$:     MOV     SAVE4,4
011500  013737  002652  000006                  MOV     SAVE6,6
011506                                          ESCAPE  TST
011506  104410                                  TRAP    C$ESCAPE
011510  000002                                  .WORD   L10052-.
011512                                  ENDTST
011512                                  L10052:
011512  104401                                  TRAP    C$ETST

011514                                          BADHEAD
                                                ;******************** TEST 2 ********************
                                                ;*VERIFY THAT RUN CAN BE CLEARED
011514                                          BADHEAD
                                                ;******************** TEST 2 ********************

011514                                  BGNTST
011514                                  T2::
011514                                          MYINT
011514  013701  002716                          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
011520  005011                                  CLR     (R1)            ;CLEAR KMCSR
011522  005002                                  CLR     R2              ;CLEAR "EXPECTED"
```

```
3318  011524  011104                        MOV     (R1),R4         ;PUT KMCSR IN 'FOUND'
3319  011526  001413                        BEQ     1$              ;BR IF CLEARED
3320  011530                                ERROR   26              ;ERROR KMCSR NOT CLEARED
3321  011546  104455                        TRAP    C$ERDF
3322  011550  000032                        .WORD   26
3323  011552  005474                        .WORD   EM26
3324  011554  007566                        .WORD   ERR26
3325  011556                        1$:
3326  011556                        ENDTST
3327  011556                        L10053:
3328  011556  104401                        TRAP    C$ETST
3329
3330  011560                                BADHEAD
3331                                        ;******************** TEST 3 ********************
3332                                        ;*UNIBUS REGISTER WORD DUAL ADDRESSING TEST
3333                                        ;*LOAD ALL REGISTERS WITH INCREMENTING PATTERN
3334                                        ;*READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING
3335                                        ;*THE SEQUENCE:
3336                                        ;*      1. CLEAR REGISTER
3337                                        ;*      2. WRITE PATTERN
3338                                        ;*      3. VERIFY PATTERN
3339                                        ;*      4. DO ALL 4 REGISTERS
3340                                        ;*      5. READ ALL BACK IF ERRORS,
3341                                        ;*         DUAL ADDRESS PROBLEM.
3342                                        ;*
3343                                        ;*         1 IN REG 0
3344                                        ;*         2 IN REG 2
3345                                        ;*         3 IN REG 4
3346                                        ;*         4 IN REG 6
3347  011560                                BADHEAD
3348                                        ;******************** TEST 3 ********************
3349
3350  011560                        BGNTST
3351  011560                        T3::
3352  011560                                MYINT
3353  011560  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
3354                                                                ;R1 CONTAINS BASE M8200,4,7 ADDRESS
3355  011564                                MSTCLR                  ;MASTER CLEAR M8200,4,7
3356  011564  004537  003156                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
3357  011570  012702  000001                MOV     #1,R2           ;START PATTERN AT 1
3358  011574                                BGNSEG
3359  011574  104404                        TRAP    C$BSEG
3360  011576  005011                1$:     CLR     (R1)            ;CLEAR REGISTER
3361  011600  010211                        MOV     R2,(R1)         ;WRITE M8200,4,7 REGISTER WITH PATTERN
3362  011602  011104                        MOV     (R1),R4         ;READ M8200,4,7 REGISTER INTO 'FOUND'
3363  011604  020204                        CMP     R2,R4           ;IS DATA CORRECT
3364  011606  001413                        BEQ     2$              ;BR IS YES
3365  011610                                ERROR   26              ;DATA ERROR
3366  011626  104455                        TRAP    C$ERDF
3367  011630  000032                        .WORD   26
3368  011632  005474                        .WORD   EM26
3369  011634  007566                        .WORD   ERR26
3370  011636                        2$:     ESCAPE   SEG
3371  011636  104410                        TRAP    C$ESCAPE
3372  011640  000014                        .WORD   10000$-.
3373  011642  005721                        TST     (R1)+           ;NEXT REGISTER
```

```
3374  011644  005202                        INC     R2              ;INCREMENT DATA PATTERN
3375  011646  022702  000005                CMP     #5,R2           ;LAST REGISTER?
3376  011652  001351                        BNE     1$              ;BR IF NO
3377  011654                                ENDSEG
3378  011654                          10000$:
3379  011654  104405                        TRAP    C$ESEG
3380  011656  013701  002716                MOV     KMCSR,R1        ;BASE M8200,4,7 ADDRESS TO R1
3381  011662  012702  000001                MOV     #1,R2           ;RESTART PATTERN AT 1
3382  011666                                BGNSEG
3383  011666  104404                        TRAP    C$BSEG
3384  011670                          3$:
3385  011670  011104                        MOV     (R1),R4         ;READ COMM. MICR-PROCESSOR FAMILY REGISTER INTO 'FOUND'
3386  011672  020204                        CMP     R2,R'           ;IS DATA CORRECT
3387  011674  001413                        BEQ     4$              ;BR IF YES
3388  011676                                ERROR   2               ;DUAL ADDRESSING ERROR
3389  011714  104455                        TRAP    C$ERDF
3390  011716  000002                        .WORD   2
3391  011720  004307                        .WORD   EM2
3392  011722  006132                        .WORD   ERR2
3393  011724                          4$:   ESCAPE  SEG
3394  011724  104410                        TRAP    C$ESCAPE
3395  011726  000014                        .WORD   10001$-.
3396  011730  005721                        TST     (R1)+           ;NEXT REGISTER
3397  011732  005202                        INC     R2              ;INCREMENT PATTERN
3398  011734  022702  000005                CMP     #5,R2           ;LAST REGISTER?
3399  011740  001353                        BNE     3$              ;BR IF NO
3400  011742                                ENDSEG
3401  011742                          10001$:
3402  011742  104405                        TRAP    C$ESEG
3403  011744                          ENDTST
3404  011744                          L10054:
3405  011744  104401                        TRAP    C$ETST
3406
3407  011746                                BADHEAD
3408                                        ;******************** TEST 4 *********************
3409                                        ;*CONTROL STATUS REGISTER WRITE/READ TEST
3410                                        ;*FLOAT A ONE THROUGH BSEL 0
3411                                        ;*CLEAR BIT0, VERIFY BIT0 WAS CLEARED
3412  011746                                BADHEAD
3413                                        ;******************** TEST 4 *********************
3414
3415  011746                          BGNTST
3416  011746                          T4::
3417  011746                                MSTCLR                  ;MASTER CLEAR M8200,4,7
3418  011746  004537  003156                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
3419  011752  005037  002624                CLR     MR0
3420  011756  012702  000001                MOV     #BIT0,R2        ;INDICATE BSEL0
3421  011762                                BGNSEG
3422  011762  104404                        TRAP    C$BSEG
3423  011764  013701  002716          1$:   MOV     KMCSR,R1        ;PUT REGISTER ADDRESS IN R1
3424  011770  010237  002636                MOV     R2,$GDDAT
3425  011774  013711  002636                MOV     $GDDAT,(R1)             ;WRITE BIT 0
3426  012000  011104                        MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
3427  012002  023704  002636                CMP     $GDDAT,R4               ;IS DATA CORRECT
3428  012006  001411                        BEQ     2$              ;BR IF YES
3429  012010                                ERROR   27,YES          ;DATA ERROR
```

```
3430  012022  104455                        TRAP    C$ERDF
3431  012024  000033                        .WORD   27
3432  012026  005525                        .WORD   EM27
3433  012030  007644                        .WORD   ERR27
3434  012032                        2$:     ESCAPE  SEG
3435  012032  104410                        TRAP    C$ESCAPE
3436  012034  000052                        .WORD   10000$-.
3437  012036  040211                3$:     BIC     R2,(R1)         ;CLEAR BSEL0
3438  012040  005037  002636                CLR     $GDDAT          ;CLEAR "EXPECTED"
3439  012044  011104                        MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
3440  012046  001413                        BEQ     4$              ;BR IF ZERO
3441  012050                                ERROR   2               ;DATA ERROR BSEL NOT CLEARED
3442  012066  104455                        TRAP    C$ERDF
3443  012070  000002                        .WORD   2
3444  012072  004307                        .WORD   EM2
3445  012074  006132                        .WORD   ERR2
3446  012076                        4$:     ESCAPE  SEG
3447  012076  104410                        TRAP    C$ESCAPE
3448  012100  000006                        .WORD   10000$-.
3449  012102  106302                        ASLB    R2
3450  012104  001327                        BNE     1$
3451  012106                                ENDSEG
3452  012106                        10000$:
3453  012106  104405                        TRAP    C$ESEG
3454  012110                        ENDTST
3455  012110                        L10055:
3456  012110  104401                        TRAP    C$ETST
3457
3458
```

```
                                        BADHEAD
                                        ;******************** TEST 5 ********************
                                        ;*CONTROL STATUS REGISTER WRITE/READ TEST
                                        ;*SET BIT9, VERIFY BIT9 WAS SET
                                        ;*CLEAR BIT9, VERIFY BIT9 WAS CLEARED
                                        BADHEAD
                                        ;******************** TEST 5 ********************

                                BGNTST
                                T5::
                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
012112  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
012116                                  BGNSEG
012116  104404                          TRAP    C$BSEG
012120  013701  002716          1$:     MOV     KMCSR,R1        ;PUT REGISTER ADDRESS IN R1
012124  012702  001000                  MOV     #BIT9,R2        ;PUT DATA IN 'EXPECTED'
012130  010211                          MOV     R2,(R1)         ;WRITE BIT 9
012132  011104                          MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
012134  020204                          CMP     R2,R4           ;IS DATA CORRECT
012136  001413                          BEQ     2$              ;BR IF YES
012140                                  ERROR   26              ;DATA ERROR
012    104455                           TRAP    C$ERDF
012100  000032                          .WORD   26
012162  005474                          .WORD   EM26
012164  007566                          .WORD   ERR26
012166                          2$:     ESCAPE  SEG
012166  104410                          TRAP    C$ESCAPE
012170  000002                          .WORD   10000$-.
012172                                  ENDSEG
012172                          10000$:
012172  104405                          TRAP    C$ESEG
012174                                  BGNSEG
012174  104404                          TRAP    C$BSEG
012176  042711  001000          3$:     BIC     #BIT9,(R1)      ;CLEAR BIT 9
012202  005002                          CLR     R2              ;CLEAR 'EXPECTED'
012204  011104                          MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
012206  001416                          BEQ     4$              ;BR IF ZERO
012210                                  ERROR   26              ;DATA ERROR BIT9 NOT CLEARED
012226  104455                          TRAP    C$ERDF
012230  000032                          .WORD   26
012232  005474                          .WORD   EM26
012234  007566                          .WORD   ERR26
012236                                  ESCAPE  SEG
012236  104410                          TRAP    C$ESCAPE
012240  000002                          .WORD   10001$-.
012242                                  ENDSEG
012242                          10001$:
012242  104405                          TRAP    C$ESEG
012244                          4$:
012244                          ENDTST
012244                          L10056:
```

```
3515  012244  104401                          TRAP    C$ETST
3516
3517  012246                                  BADHEAD
3518                                          ;******************** TEST 6 ********************
3519                                          ;*CONTROL STATUS REGISTER WRITE/READ TEST
3520                                          ;*SET BIT11, VERIFY BIT11 WAS SET
3521                                          ;*CLEAR BIT11, VERIFY BIT11 WAS CLEARED
3522  012246                                  BADHEAD
3523                                          ;******************** TEST 6 ********************
3524
3525  012246                          BGNTST
3526  012246                          T6::
3527  012246                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
3528  012246  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
3529  012252                                  BGNSEG
3530  012252  104404                          TRAP    C$BSEG
3531  012254  013701  002716          1$:     MOV     KMCSR,R1        ;PUT REGISTER ADDRESS IN R1
3532  012260  012702  004000                  MO      #BIT11,R2       ;PUT DATA IN "EXPECTED"
3533  012264  010211                          MOV     R2,(R1)         ;WRITE BIT 11
3534  012266  011104                          MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
3535  012270  020204                          CMP     R2,R4           ;IS DATA CORRECT
3536  012272  001413                          BEQ     2$              ;BR IF YES
3537  012274                                  ERROR   26              ;DATA ERROR
3538  012312  104455                          TRAP    C$ERDF
3539  012314  000032                          .WORD   26
3540  012316  005474                          .WORD   EM26
3541  012320  007566                          .WORD   ERR26
3542  012322                          2$:     ESCAPE  SEG
3543  012322  104410                          TRAP    C$ESCAPE
3544  012324  000002                          .WORD   10000$-.
3545  012326                                  ENDSEG
3546  012326                          10000$:
3547  012326  104405                          TRAP    C$ESEG
3548  012330                                  BGNSEG
3549  012330  104404                          TRAP    C$BSEG
3550  012332  042711  004000          3$:     BIC     #BIT11,(R1)     ;CLEAR BIT 11
3551  012336  005002                          CLR     R2              ;CLEAR "EXPECTED"
3552  012340  011104                          MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
3553  012342  001414                          BEQ     4$              ;BR IF ZERO
3554  012344                                  ERROR   26              ;DATA ERROR BIT11 NOT CLEARED
3555  012362  104455                          TRAP    C$ERDF
3556  012364  000032                          .WORD   26
3557  012366  005474                          .WORD   EM26
3558  012370  007566                          .WORD   ERR26
3559  012372                                  ENDSEG
3560  012372                          10001$:
3561  012372  104405                          TRAP    C$ESEG
3562  012374                          4$:
3563  012374                          ENDTST
3564  012374                          L10057:
3565  012374  104401                          TRAP    C$ETST
3566
3567  012376                                  BADHEAD
3568                                          ;******************** TEST 7 ********************
3569                                          ;*CONTROL STATUS REGISTER WRITE/READ TEST
3570                                          ;*SET BIT12, VERIFY BIT12 WAS SET
```

```
3571                                                    ;*CLEAR BIT12, VERIFY BIT12 WAS CLEARED
3572  012376                                            BADHEAD
3573                                                    ;******************** TEST 7 ********************
3574
3575  012376                                    BGNTST
3576  012376                                    T7::
3577  012376                                            MSTCLR                  ;MASTER CLEAR M8200,4,7
3578  012376  004537  003156                           JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
3579  012402                                            BGNSEG
3580  012402  104404                                   TRAP    C$BSEG
3581  012404  013701  002716                   1$:     MOV     KMCSR,R1        ;PUT REGISTER ADDRESS IN R1
3582  012410  012702  010000                           MOV     #BIT12,R2       ;PUT DATA IN "EXPECTED"
3583  012414  010211                                   MOV     R2,(R1)         ;WRITE BIT 12
3584  012416  011104                                   MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
3585  012420  020204                                   CMP     R2,R4           ;IS DATA CORRECT
3586  012422  001413                                   BEQ     2$              ;BR IF YES
3587  012424                                            ERROR   26              ;DATA ERROR
3588  012442  104455                                   TRAP    C$ERDF
3589  012444  000032                                   .WORD   26
3590  012446  005474                                   .WORD   EM26
3591  012450  007566                                   .WORD   ERR26
3592  012452                                    2$:     ESCAPE  SEG
3593  012452  104410                                   TRAP    C$ESCAPE
3594  012454  000002                                   .WORD   10000$-.
3595  012456                                            ENDSEG
3596  012456                                    10000$:
3597  012456  104405                                   TRAP    C$ESEG
3598  012460                                            BGNSEG
3599  012460  104404                                   TRAP    C$BSEG
3600  012462  042711  010000                   3$:     BIC     #BIT12,(R1)     ;CLEAR BIT 12
3601  012466  005002                                   CLR     R2              ;CLEAR "EXPECTED"
3602  012470  011104                                   MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
3603  012472  001414                                   BEQ     4$              ;BR IF ZERO
3604  012474                                            ERROR   26              ;DATA ERROR BIT12 NOT CLEARED
3605  012512  104455                                   TRAP    C$ERDF
3606  012514  000032                                   .WORD   26
3607  012516  005474                                   .WORD   EM26
3608  012520  007566                                   .WORD   ERR26
3609  012522                                            ENDSEG
3610  012522                                    10001$:
3611  012522  104405                                   TRAP    C$ESEG
3612  012524                                    4$:
3613  012524                                    ENDTST
3614  012524                                    L10060:
3615  012524  104401                                   TRAP    C$ETST
3616
3617  012526                                            BADHEAD
3618                                                    ;******************** TEST 8 ********************
3619                                                    ;*CONTROL OUT REGISTER WRITE/READ TEST
3620                                                    ;*FLOAT A ONE THROUGH SEL2
3621  012526                                            BADHEAD
3622                                                    ;******************** TEST 8 ********************
3623
3624  012526                                    BGNTST
3625  012526                                    T8::
3626  012526                                            MSTCLR                  ;MASTER CLEAR M8200,4,7
```

```
3627  012526  004537  002156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
3628  012532  012737  000002  002624          MOV     #2,MR0
3629  012540  012702  000001                  MOV     #1,R2
3630  012544                                  BGNSEG
3631  012544  104404                         TRAP    C$BSEG
3632
3633  012546  013701  002722          1$:     MOV     KMCTL,R1        ;PUT REGISTER ADDRESS IN R1
3634  012552  010237  002636                  MOV     R2,$GDDAT       ;PUT DATA IN "EXPECTED"
3635  012556  013711  002636                  MOV     $GDDAT,(R1)             ;WRITE BIT 0
3636  012562  011104                          MOV     (R1),R4         ;READ CONTROL OUT REGISTER
3637  012564  023704  002636                  CMP     $GDDAT,R4               ;IS DATA CORRECT
3638  012570  001411                          BEQ     2$              ;BR IF YES
3639  012572                                  ERROR   27,YES          ;DATA ERROR
3640  012604  104455                         TRAP    C$ERDF
3641  012606  000033                         .WORD   27
3642  012610  005525                         .WORD   EM27
3643  012612  007644                         .WORD   ERR27
3644  012614                          2$:     ESCAPE  SEG
3645  012614  104410                         TRAP    C$ESCAPE
3646  012616  000046                         .WORD   10000$-.
3647  012620  040211                  3$:     BIC     R2,(R1) ;CLEAR BIT
3648  012622  005037  002636                  CLR     $GDDAT          ;CLEAR "EXPECTED"
3649  012626  011104                          MOV     (R1),R4         ;READ CONTROL OUT REGISTER
3650  012630  001411                          BEQ     4$              ;BR IF ZERO
3651  012632                                  ERROR   27,YES          ;DATA ERROR BIT0 NOT CLEARED
3652  012644  104455                         TRAP    C$ERDF
3653  012646  000033                         .WORD   27
3654  012650  005525                         .WORD   EM27
3655  012652  007644                         .WORD   ERR27
3656  012654                          4$:     ESCAPEE SEG
3657  012654  104410                         TRAP    C$ESCAPE
3658  012656  000006                         .WORD   10000$-.
3659  012660  006302                         ASL     R2
3660  012662  001331                         BNE     1$
3661  012664                                  ENDSEG
3662  012664                          10000$:
3663  012664  104405                         TRAP    C$ESEG
3664  012666                          ENDTST
3665  012666                          L10061:
3666  012666  104401                         TRAP    C$ETST
3667
3668
3669
3670
3671
3672
3673  012670                                  BADHEAD
3674                                          ;******************** TEST 9 ********************
3675                                          ;*PORT4 REGISTER WRITE/READ TEST
3676                                          ;*FLOAT A ONE THROUGH PORT4 REGISTER
3677                                          ;*FLOAT A ZERO THROUGH PORT4 REGISTER
3678  012670                                  BADHEAD
3679                                          ;******************** TEST 9 ********************
3680
3681
3682  012670                          BGNTST
```

```
3683  012670                          T9::
3684  012670  012737  000004  002624          MOV     #4,MR0
3685  012676                          MSTCLR                  ;MASTER CLEAR M8200,4,7
3686  012676  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
3687  012702  013701  002724                  MOV     KMP04,R1        ;PUT REGISTER ADDRESS IN R1
3688  012706  012702  000001                  MOV     #1,R2            ;START WITH BIT0
3689  012712                                  BGNSEG
3690  012712  104404                          TRAP    C$BSEG
3691  012714                          64$:
3692  012714  010211                          MOV     R2,(R1)         ;WRITE PORT4 REGISTER
3693  012716  011104                          MOV     (R1),R4         ;READ PORT4 REGISTER
3694  012720  020204                          CMP     R2,R4           ;COMPARE EXPECTED AND FOUND
3695  012722  001413                          BEQ     65$             ;BR IF OK
3696  012724                                  ERROR   27              ;WRITE/READ ERROR
3697  012742  104455                          TRAP    C$ERDF
3698  012744  000033                          .WORD   27
3699  012746  005525                          .WORD   EM27
3700  012750  007644                          .WORD   ERR27
3701  012752                          65$:    ESCAPE  SEG
3702  012752  104410                          TRAP    C$ESCAPE
3703  012754  000010                          .WORD   10000$-.
3704  012756  000241                          CLC                     ;CLEAR CARRY
3705  012760  006102                          ROL     R2              ;SHIFT TO NEXT BIT
3706  012762  001354                          BNE     64$             ;BR IF NOT DONE YET?
3707  012764                                  ENDSEG
3708  012764                          10000$:
3709  012764  104405                          TRAP    C$ESEG
3710  012766  012702  000001                  MOV     #1,R2           ;START WITH BIT0
3711  012772                                  BGNSEG
3712  012772  104404                          TRAP    C$BSEG
3713  012774                          66$:
3714  012774  005102                          COM     R2              ;CHANGE TO A FLOATING ZERO
3715  012776  010211                          MOV     R2,(R1)         ;WRITE PORT4 REGISTER
3716  013000  011104                          MOV     (R1),R4
3717  013002  020204                          CMP     R2,R4           ;COMPARE EXPECTED AND FOUND
3718  013004  001413                          BEQ     67$             ;BR IF OK
3719  013006                                  ERROR   27              ;WRITE/READ ERROR
3720  013024  104455                          TRAP    C$ERDF
3721  013026  000033                          .WORD   27
3722  013030  005525                          .WORD   EM27
3723  013032  007644                          .WORD   ERR27
3724  013034                          67$:    ESCAPE  SEG
3725  013034  104410                          TRAP    C$ESCAPE
3726  013036  000012                          .WORD   10001$-.
3727  013040  005102                          COM     R2              ;CHANGE BACK TO A FLOATING ONE
3728  013042  000241                          CLC                     ;CLEAR CARRY
3729  013044  006102                          ROL     R2              ;SHIFT TO NEXT BIT
3730  013046  001352                          BNE     66$             ;BR IF NOT DONE YET?
3731  013050                                  ENDSEG
3732  013050                          10001$:
3733  013050  104405                          TRAP    C$ESEG
3734  013052                          ENDTST
3735  013052                          L10062:
3736  013052  104401                          TRAP    C$ETST
3737
3738  013054                                  BADHEAD
```

```
3739                                            ;******************** TEST 10 ********************
3740                                            ;*PORT6 REGISTER WRITE/READ TEST
3741                                            ;*FLOAT A ONE THROUGH PORT6 REGISTER
3742                                            ;*FLOAT A ZERO THROUGH PORT6 REGISTER
3743  013054                                    BADHEAD
3744                                            ;******************** TEST 10 ********************
3745
3746  013054                                BGNTST
3747  013054                                T10::
3748  013054  012737  000006  002624               MOV     #6,MR0
3749  013062                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
3750  013062  004537  003156                       JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
3751  013066  013701  002726                       MOV     KMP06,R1        ;PUT REGISTER ADDRESS IN R1
3752  013072  012702  000001                       MOV     #1,R2            ;START WITH BIT0
3753  013076                                        BGNSEG
3754  013076  104404                               TRAP    C$BSEG
3755  013100                                64$:
3756  013100  010211                               MOV     R2,(R1)         ;WRITE PORT6 REGISTER
3757  013102  011104                               MOV     (R1),R4         ;READ PORT6 REGISTER
3758  013104  020204                               CMP     R2,R4           ;COMPARE EXPECTED AND FOUND
3759  013106  001413                               BEQ     65$             ;BR IF OK
3760  013110                                        ERROR   27              ;WRITE/READ ERROR
3761  013126  104455                               TRAP    C$ERDF
3762  013130  000033                               .WORD   27
3763  013132  005525                               .WORD   EM27
3764  013134  007644                               .WORD   ERR27
3765  013136                                65$:    ESCAPE  SEG
3766  013136  104410                               TRAP    C$ESCAPE
3767  013140  000010                               .WORD   10000$-.
3768  013142  000241                               CLC                     ;CLEAR CARRY
3769  013144  006105                               ROL     R5              ;SHIFT TO NEXT BIT
3770  013146  001354                               BNE     64$             ;BR IF NOT DONE YET?
3771  013150                                        ENDSEG
3772  013150                                10000$:
3773  013150  104405                               TRAP    C$ESEG
3774  013152  012702  000001                       MOV     #1,R2           ;START WITH BIT0
3775  013156                                        BGNSEG
3776  013156  104404                               TRAP    C$BSEG
3777  013160                                66$:
3778  013160  005102                               COM     R2              ;CHANGE TO A FLOATING ZERO
3779
3780  013162  010211                               MOV     R2,(R1)         ;WRITE PORT6 REGISTER
3781  013164  011104                               MOV     (R1),R4         ;READ PORT6 REGISTER
3782  013166  020204                               CMP     R2,R4           ;COMPARE EXPECTED AND FOUND
3783  013170  001413                               BEQ     67$             ;BR IF OK
3784  013172                                        ERROR   27              ;WRITE/READ ERROR
3785  013210  104455                               TRAP    C$ERDF
3786  013212  000033                               .WORD   27
3787  013214  005525                               .WORD   EM27
3788  013216  007644                               .WORD   ERR27
3789  013220                                67$:    ESCAPE  SEG
3790  013220  104410                               TRAP    C$ESCAPE
3791  013222  000012                               .WORD   10001$-.
3792  013224  005102                               COM     R2              ;CHANGE BACK TO A FLOATING ONE
3793  013226  000241                               CLC                     ;CLEAR CARRY
3794  013230  006102                               ROL     R2              ;SHIFT TO NEXT BIT
```

```
3795  013232  001352                        BNE     66$             ;BR IF NOT DONE YET?
3796  013234                                ENDSEG
3797  013234                          10001$:
3798  013234  104405                        TRAP    C$ESEG
3799  013236                          ENDTST
3800  013236                          L10063:
3801  013236  104401                        TRAP    C$ETST
3802
3803  013240                                BADHEAD
3804                                        ;******************** TEST 11 *********************
3805                                        ;*UNIBUS REGISTER BYTE DUAL ADDRESSING TEST
3806                                        ;*LOAD ALL REGISTERS WITH INCREMENTING PATTERN
3807                                        ;*READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING
3808  013240                                BADHEAD
3809                                        ;******************** TEST 11 *********************
3810
3811  013240                          BGNTST
3812  013240                          T11::
3813  013240                                MYINT
3814  013240  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
3815  013244                                MSTCLR                  ;MASTER CLEAR M8200,4,7
3816  013244  004537  003156                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
3817  013250  012702  000001                MOV     #1,R2           ;START PATTERN AT 1
3818  013254                                BGNSEG
3819  013254  104404                        TRAP    C$BSEG
3820  013256  105011                  1$:   CLRB    (R1)            ;CLEAR REGISTER
3821  013260  110211                        MOVB    R2,(R1)         ;WRITE M8200,4,7 REGISTER WITH PATTERN
3822  013262  111104                        MOVB    (R1),R4         ;READ M8200,4,7 REGISTER INTO "FOUND"
3823  013264  120204                        CMPB    R2,R4           ;IS DATA CORRECT
3824  013266  001413                        BEQ     2$              ;BR IF YES
3825  013270                                ERROR   2               ;DATA ERROR
3826  013306  104455                        TRAP    C$ERDF
3827  013310  000002                        .WORD   2
3828  013312  004307                        .WORD   EM2
3829  013314  006132                        .WORD   ERR2
3830  013316                          2$:   ESCAPE  SEG
3831  013316  104410                        TRAP    C$ESCAPE
3832  013320  000024                        .WORD   10000$-.
3833  013322  105721                        TSTB    (R1)+           ;NEXT REGISTER
3834  013324  005202                        INC     R2              ;INCREMENT DATA PATTERN
3835  013326  022702  000011                CMP     #11,R2          ;LAST REGISTER?
3836  013332  001351                        BNE     1$              ;BR IF NO
3837  013334  013701  002716                MOV     KMCSR,R1        ;BASE M8200,4,7 ADDRESS TO R1
3838  013340  012702  000001                MOV     #1,R2           ;RESTART PATTERN AT 1
3839  013344                                ENDSEG
3840  013344                          10000$:
3841  013344  104405                        TRAP    C$ESEG
3842  013346                                BGNSEG
3843  013346  104404                        TRAP    C$BSEG
3844  013350                          3$:
3845  013350  111104                        MOVB    (R1),R4         ;READ COMM.MICRO-PROCESSOR FAMILY REGISTER INTO "FOUND"
3846  013352  120204                        CMPB    R2,R4           ;IS DATA CORRECT
3847  013354  001413                        BEQ     4$              ;BR IF YES
3848  013356                                ERROR   2               ;DUAL ADDRESSING ERROR
3849  013374  104455                        TRAP    C$ERDF
3850  013376  000002                        .WORD   2
```

```
3851  013400  004307                        .WORD   EM2
3852  013402  006132                        .WORD   ERR2
3853  013404                        4$:     ESCAPE  SEG
3854  013404  104410                        TRAP    C$ESCAPE
3855  013406  000014                        .WORD   10001$-.
3856  013410  105721                        TSTB    (R1)+           ;NEXT REGISTER
3857  013412  005202                        INC     R2              ;INCREMENT PATTERN
3858  013414  022702  000011                CMP     #11,R2          ;LAST REGISTER?
3859  013420  001353                        BNE     3$              ;BR IF NO
3860  013422                                ENDSEG
3861  013422                        10001$:
3862  013422  104405                        TRAP    C$ESEG
3863  013424                        ENDTST
3864  013424                        L10064:
3865  013424  104401                        TRAP    C$ETST
3866
3867  013426                                BADHEAD
3868                                        ;********************* TEST 12 *********************
3869                                        ;*MAINTENANCE INSTRUCTION REGISTER TEST
3870                                        ;*VERIFY THAT THE MAINT IR CAN BE WRITTEN TO ALL ZEROS'
3871                                        ;*AND ALL ONES'. VERIFY THAT IT IS CLEARED ON A BUS RESET.
3872  013426                                BADHEAD
3873                                        ;********************* TEST 12 *********************
3874
3875  013426                        BGNTST
3876  013426                        T12::
3877                                                                ;R1 CONTAINS BASE M8200,4,7 ADDRESS
3878  013426                                MSTCLR                  ;MASTER CLEAR M8200,4,7
3879  013426  004537  003156                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
3880  013432                                MYINT
3881  013432  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
3882  013436                                BGNSEG
3883  013436  104404                        TRAP    C$BSEG
3884  013440  012711  003000                MOV     #BIT9!BIT10,(R1) ;SEL6 IS NOW THE IR
3885  013444  005002                        CLR     R2              ;PUT "EXPECTED" IN $GDDAT
3886  013446  010261  000006        1$:     MOV     R2,6(R1)        ;CLEAR THE IR
3887  013452  016104  000006                MOV     6(R1),R4        ;READ THE IR
3888  013456  020204                        CMP     R2,R4           ;IS IT CLEARED?
3889  013460  001413                        BEQ     2$              ;BR IF YES
3890  013462                                ERROR   26              ;ERROR IR IS NOT CLEAR
3891  013500  104455                        TRAP    C$ERDF
3892  013502  000032                        .WORD   26
3893  013504  005474                        .WORD   EM26
3894  013506  007566                        .WORD   ERR26
3895  013510                        2$:     ESCAPE  SEG
3896  013510  104410                        TRAP    C$ESCAPE
3897  013512  000002                        .WORD   10000$-.
3898  013514                                ENDSEG
3899  013514                        10000$:
3900  013514  104405                        TRAP    C$ESEG
3901  013516  012702  177777                MOV     #-1,R2          ;PUT "EXPECTED" IN $GDDAT
3902  013522                                BGNSEG
3903  013522  104404                        TRAP    C$BSEG
3904  013524  010261  000006        3$:     MOV     R2,6(R1)        ;WRITE ALL ONES TO THE IR
3905  013530  016104  000006                MOV     6(R1),R4        ;READ THE IR
3906  013534  020204                        CMP     R2,R4           ;IS IT ALL ONES?
```

```
 3907  013536  001413                    BEQ     4$              ;BR IF YES
 3908  013540                            ERROR   26              ;ERROR IR IS NOT = ALL ONES
 3909  013556  104455                    TRAP    C$ERDF
 3910  013560  000032                    .WORD   26
 3911  013562  005474                    .WORD   EM26
 3912  013564  007566                    .WORD   ERR26
 3913  013566                    4$:     ESCAPE  SEG
 3914  013566  104410                    TRAP    C$ESCAPE
 3915  013570  000002                    .WORD   10001$-.
 3916  013572                            ENDSEG
 3917  013572                    10001$:
 3918  013572  104405                    TRAP    C$ESEG
 3919  013574                    ENDTST
 3920  013574                    L10065:
 3921  013574  104401                    TRAP    C$ETST
 3922
 3923
```

```
3924
3925  013576                                  BADHEAD
3926                                          ;******************** TEST 13 ********************
3927                                          ;*MICRO PROCESSOR TEST
3928                                          ;*LOAD KMP06 WITH A MICRO-PROCESSOR INSTRUCTION, CLOCK IT
3929                                          ;*VERIFY INSTRUCTION EXECUTED PROPERLY
3930                                          ;*INSTRUCTION SHOULD MOVE IBUS*4 TO IBUS*5, IBUS*4 IS ALL 1'S
3931                                          ;*AND IBUS*5 IS ALL 0'S. RESULT SHOULD BE ALL 1'S IN SEL4
3932  013576                                  BADHEAD
3933                                          ;******************** TEST 13 ********************
3934
3935  013576                          BGNTST
3936  013576                          T13::
3937  013576                                  MYINT
3938  013576  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
3939  013602                                  MSTCLR
3940  013602  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
3941  013606  012761  000377  000004          MOV     #377,4(R1)      ;PORT4 HI BYTE=1'S
3942  013614  012711  001000                  MOV     #BIT9,(R1)              ;SET ROMI
3943  013620  012761  121105  000006          MOV     #121105,6(R1)   ;INSTR TO PORT 6.
3944  013626  052711  001400                  BIS     #BIT8!BIT9,(R1) ;CLK INSTR.
3945  013632  000240                          NOP
3946  013634  012702  177777                  MOV     #-1,R2          ;EXPECT ALL ONES.
3947  013640  016104  000004                  MOV     4(R1),R4        ;READ FOUND. MOVB CHANGED TO MOV 30-AUG-82
3948  013644  020204                          CMP     R2,R4   ;DATA CORRECT?
3949  013646  001413                          BEQ     1$
3950  013650                                  ERROR   28
3951  013666  104455                          TRAP    C$ERDF
3952  013670  000034                          .WORD   28
3953  013672  005554                          .WORD   EM28
3954  013674  007726                          .WORD   ERR28
3955
3956  013676                          1$:     ESCAPE  TST
3957  013676  104410                          TRAP    C$ESCAPE
3958  013700  000002                          .WORD   L10066-.
3959
3960  013702                          ENDTST
3961  013702                          L10066:
3962  013702  104401                          TRAP    C$ETST
3963
3964  013704                                  BADHEAD
3965                                          ;******************** TEST 14 ********************
3966                                          ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
3967                                          ;*FLOAT A 1 THOUGH IBUS* REGISTER 0
3968                                          ;*FLOAT A 0 THROUGH IBUS* REGISTER 0
3969  013704                                  BADHEAD
3970                                          ;******************** TEST 14 ********************
3971
3972  013704                          BGNTST
3973  013704                          T14::
3974  013704                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
3975  013704  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
3976  013710  012737  000000  002624          MOV     #0,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
3977  013716  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
3978
3979  013722                                  MYINT
```

```
3980  013722  013701  002716          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
3981  013726                          BGNSEG
3982  013726  104404                  TRAP    C$BSEG
3983  013730                  64$:
3984  013730  010561  000004          MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
3985  013734                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
3986  013734  004537  003244          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
3987  013740  121100                  121100                  ;MOV DATA TO IBUS* REGISTER 0
3988  013742                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
3989  013742  004537  003244          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
3990  013746  121005                  121005                  ;READ FROM IBU* REGITER 0
3991  013750  116104  000005          MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
3992  013754  120504                  CMPB    R5,R4           ;DATA CORRECT?
3993  013756  001414                  BEQ     65$             ;BR IF YES
3994  013760                          BERROR  27              ;ERROR
3995  014000  104455                  TRAP    C$ERDF
3996  014002  000033                  .WORD   27
3997  014004  005525                  .WORD   EM27
3998  014006  007644                  .WORD   ERR27
3999  014010                  65$:    ESCAPE  SEG
4000  014010  104410                  TRAP    C$ESCAPE
4001  014012  000010                  .WORD   10000$-.
4002  014014  000241                  CLC                     ;CLEAR CARRY
4003  014016  106105                  ROLB    R5              ;SHIFT BIT IN R5
4004  014020  001343                  BNE     64$             ;IF R2=0 THEN DONE
4005  014022                          ENDSEG
4006  014022                  10000$:
4007  014022  104405                  TRAP    C$ESEG
4008  014024  012705  000001          MOV     #1,R5           ;START WITH BIT 0
4009                          ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
4010  014030                          BGNSEG
4011  014030  104404                  TRAP    C$BSEG
4012  014032                  67$:
4013  014032  005105                  COM     R5
4014  014034  010561  000004          MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
4015  014040                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4016  014040  004537  003244          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4017  014044  121100                  121100                  ;MOV DATA TO IBUS* REGISTER 0
4018  014046                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4019  014046  004537  003244          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4020  014052  121005                  121005                  ;READ FROM IBUS* REGISTER 0
4021  014054  116104  000005          MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
4022  014060  120504                  CMPB    R5,R4           ;DATA CORRECT?
4023  014062  001414                  BEQ     68$             ;BR IF YES
4024  014064                          BERROR  27              ;ERROR
4025  014104  104455                  TRAP    C$ERDF
4026  014106  000033                  .WORD   27
4027  014110  005525                  .WORD   EM27
4028  014112  007644                  .WORD   ERR27
4029  014114                  68$:    ESCAPE  SEG
4030  014114  104410                  TRAP    C$ESCAPE
4031  014116  000012                  .WORD   10001$-.
4032  014120  005105                  COM     R5              ;CHANGE TO FLOATING 1
4033  014122  000241                  CLC                     ;CLEAR CARRY
4034  014124  106105                  ROLB    R5              ;SHIFT BIT IN R5
4035  014126  001341                  BNE     67$             ;IF R2=0 THEN DONE
```

```
4036  014130                              ENDSEG
4037  014130                        10001$:
4038  014130  104405                      TRAP    C$ESEG
4039  014132                        ENDTST
4040  014132                        L10067:
4041  014132  104401                      TRAP    C$ETST
4042
4043  014134                              BADHEAD
4044                                      ;******************** TEST 15 ********************
4045                                      ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
4046                                      ;*FLOAT A 1 THOUGH IBUS* REGISTER 2
4047                                      ;*FLOAT A 0 THROUGH IBUS* REGISTER 2
4048  014134                              BADHEAD
4049                                      ;******************** TEST 15 ********************
4050
4051  014134                        BGNTST
4052  014134                        T15::
4053  014134                              MSTCLR                  ;MASTER CLEAR M8200,4,7
4054  014134  004537  003156             JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
4055  014140  012737  000002  002624     MOV     #2,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
4056  014146  012705  000001             MOV     #1,R5           ;START WITH BIT 0
4057  014152                              MYINT
4058  014152  013701  002716             MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
4059  014156                              BGNSEG
4060  014156  104404                      TRAP    C$BSEG
4061  014160                        64$:
4062  014160  010561  000004             MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
4063  014164                              ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4064  014164  004537  003244             JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4065  014170  121102                      121100!2                ;MOV DATA TO IBUS* REGISTER 0
4066  014172                              ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4067  014172  004537  003244             JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4068  014176  121045                      121005!<2*20>           ;READ FROM IBUS* REGISTER 2
4069  014200  116104  000005             MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
4070  014204  120504                      CMPB    R5,R4           ;DATA CORRECT?
4071  014206  001414                      BEQ     65$             ;BR IF YES
4072  014210                              BERROR  27              ;ERROR
4073  014230  104455                      TRAP    C$ERDF
4074  014232  000033                      .WORD   27
4075  014234  005525                      .WORD   EM27
4076  014236  007644                      .WORD   ERR27
4077  014240                        65$:  ESCAPE  SEG
4078  014240  104410                      TRAP    C$ESCAPE
4079  014242  000010                      .WORD   10000$-.
4080  014244  000241                      CLC                     ;CLEAR CARRY
4081  014246  106105                      ROLB    R5              ;SHIFT BIT IN R2
4082  014250  001343                      BNE     64$             ;IF R2=0 THEN DONE
4083  014252                              ENDSEG
4084  014252                        10000$:
4085  014252  104405                      TRAP    C$ESEG
4086  014254  012705  000001             MOV     #1,R5           ;START WITH BIT 0
4087                                ;69$: COM     R5              ;CHANGE TO FLOATING ZERO
4088  014260                              BGNSEG
4089  014260  104404                      TRAP    C$BSEG
4090  014262                        67$:
4091  014262  005105                      COM     R5
```

```
4092  014264  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
4093  014270                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4094  014270  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4095  014274  121102                          121100!2                ;MOV DATA TO IBUS* REGISTER 2
4096  014276                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4097  014276  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4098  014302  121045                          121005!<2*20>           ;READ FROM IBUS* REGISTER 2
4099  014304  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
4100  014310  120504                          CMPB    R5,R4           ;DATA CORRECT?
4101  014312  001414                          BEQ     68$             ;BR IF YES
4102  014314                                  BERROR  27              ;ERROR
4103  014334  104455                          TRAP    C$ERDF
4104  014336  000033                          .WORD   27
4105  014340  005525                          .WORD   EM27
4106  014342  007644                          .WORD   ERR27
4107  014344                          68$:    ESCAPE  SEG
4108  014344  104410                          TRAP    C$ESCAPE
4109  014346  000012                          .WORD   10001$-.
4110  014350  005105                          COM     R5              ;CHANGE TO FLOATING 1
4111  014352  000241                          CLC                     ;CLEAR CARRY
4112  014354  106105                          ROLB    R5              ;SHIFT BIT IN R2
4113  014356  001341                          BNE     67$             ;IF R2=0 THEN DONE
4114  014360                                  ENDSEG
4115  014360                          10001$:
4116  014360  104405                          TRAP    C$ESEG
4117  014362                          ENDTST
4118  014362                          L10070:
4119  014362  104401                          TRAP    C$ETST
4120
4121  014364                                  BADHEAD
4122                                          ;*********************** TEST 16 *********************
4123                                          ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
4124                                          ;*FLOAT A 1 THOUGH IBUS* REGISTER 4
4125                                          ;*FLOAT A 0 THROUGH IBUS* REGISTER 4
4126  014364                                  BADHEAD
4127                                          ;********************* TEST 16 *********************
4128
4129  014364                          BGNTST
4130  014364                          T16::
4131  014364                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
4132  014364  004537  003156                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
4133  014370  012737  000004  002624          MOV     #4,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
4134  014376  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
4135  014402                                  MYINT
4136  014402  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
4137  014406                                  BGNSEG
4138  014406  104404                          TRAP    C$BSEG
4139  014410                          64$:
4140  014410  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
4141  014414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4142  014414  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4143  014420  121104                          121100!4                ;MOV DATA TO IBUS* REGISTER 4
4144  014422                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4145  014422  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4146  014426  121105                          121005!<4*20>           ;READ FROM IBUS* REGISTER 4
4147  014430  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
```

```
 4148  014434  120504                        CMPB    R5,R4           ;DATA CORRECT?
 4149  014436  001414                        BEQ     65$             ;BR IF YES
 4150  014440                                BERROR  27              ;ERROR
 4151  014460  104455                        TRAP    C$ERDF
 4152  014462  000033                        .WORD   27
 4153  014464  005525                        .WORD   EM27
 4154  014466  007644                        .WORD   ERR27
 4155  014470                          65$:  ESCAPE  SEG
 4156  014470  104410                        TRAP    C$ESCAPE
 4157  014472  000010                        .WORD   10000$-.
 4158  014474  000241                        CLC                     ;CLEAR CARRY
 4159  014476  106105                        ROLB    R5              ;SHIFT BIT IN R2
 4160  014500  001343                        BNE     64$             ;IF R2=0 THEN DONE
 4161  014502                                ENDSEG
 4162  014502                          10000$:
 4163  014502  104405                        TRAP    C$ESEG
 4164  014504  012705  000001                MOV     #1,R5           ;START WITH BIT 0
 4165                            ;69$:       COM     R5              ,CHANGE TO FLOATING ZERO
 4166  014510                                BGNSEG
 4167  014510  104404                        TRAP    C$BSEG
 4168  014512                          67$:
 4169  014512  005105                        COM     R5
 4170  014514  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4171  014520                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4172  014520  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 4173  014524  121104                        121100!4                ;MOV DATA TO IBUS* REGISTER 4
 4174  014526                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4175  014526  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 4176  014532  121105                        121005!<4*20>           ;READ FROM IBUS* REGISTER 4
 4177  014534  116104  000005                MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 4178  014540  120504                        CMPB    R5,R4           ;DATA CORRECT?
 4179  014542  001414                        BEQ     68$             ;BR IF YES
 4180  014544                                BERROR  27              ;ERROR
 4181  014564  104455                        TRAP    C$ERDF
 4182  014566  000033                        .WORD   27
 4183  014570  005525                        .WORD   EM27
 4184  014572  007644                        .WORD   ERR27
 4185  014574                          68$:  ESCAPE  SEG
 4186  014574  104410                        TRAP    C$ESCAPE
 4187  014576  000012                        .WORD   10001$-.
 4188  014600  005105                        COM     R5              ;CHANGE TO FLOATING 1
 4189  014602  000241                        CLC                     ;CLEAR CARRY
 4190  014604  106105                        ROLB    R5              ;SHIFT BIT IN R2
 4191  014606  001341                        BNE     67$             ;IF R2=0 THEN DONE
 4192  014610                                ENDSEG
 4193  014610                          10001$:
 4194  014610  104405                        TRAP    C$ESEG
 4195  014612                          ENDTST
 4196  014612                          L10071:
 4197  014612  104401                        TRAP    C$ETST
 4198
 4199  014614                                BADHEAD
 4200                                  ;******************** TEST 17 ********************
 4201                                  ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
 4202                                  ;*FLOAT A 1 THOUGH IBUS* REGISTER 5
 4203                                  ;*FLOAT A 0 THROUGH IBUS* REGISTER 5
```

```
4204  014614                                    BADHEAD
4205                                            ;******************** TEST 17 ********************
4206
4207  014614                              BGNTST
4208  014614                              T17::
4209  014614                                    MSTCLR                  ;MASTER CLEAR M8200,4,7
4210  014614  004537  003156                    JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
4211  014620  012737  000005  002624            MOV     #5,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
4212  014626  012705  000001                    MOV     #1,R5           ;START WITH BIT 0
4213  014632                                    MYINT
4214  014632  013701  002716                    MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
4215  014636                                    BGNSEG
4216  014636  104404                            TRAP    C$BSEG
4217  014640                              64$:
4218  014640  010561  000004                    MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
4219  014644                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4220  014644  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4221  014650  121105                            121100!5                ;MOV DATA TO IBUS* REGISTER 5
4222  014652                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4223  014652  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4224  014656  121125                            121005!<5*20>           ;READ FROM IBUS* REGISTER 5
4225  014660  116104  000005                    MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
4226  014664  120504                            CMPB    R5,R4           ;DATA CORRECT?
4227  014666  001414                            BEQ     65$             ;BR IF YES
4228  014670                                    BERROR  27              ;ERROR
4229  014710  104455                            TRAP    C$ERDF
4230  014712  000033                            .WORD   27
4231  014714  005525                            .WORD   EM27
4232  014716  007644                            .WORD   ERR27
4233  014720                              65$:  ESCAPE  SEG
4234  014720  104410                            TRAP    C$ESCAPE
4235  014722  000010                            .WORD   10000$-.
4236  014724  000241                            CLC                     ;CLEAR CARRY
4237  014726  106105                            ROLB    R5              ;SHIFT BIT IN R5
4238  014730  001343                            BNE     64$             ;IF R5=0 THEN DONE
4239  014732                                    ENDSEG
4240  014732                              10000$:
4241  014732  104405                            TRAP    C$ESEG
4242  014734  012705  000001                    MOV     #1,R5           ;START WITH BIT 0
4243                                      ;69$: COM     R5              ;CHANGE TO FLOATING ZERO
4244  014740                                    BGNSEG
4245  014740  104404                            TRAP    C$BSEG
4246  014742                              67$:
4247  014742  005105                            COM     R5
4248  014744  010561  000004                    MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
4249  014750                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4250  014750  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4251  014754  121105                            121100!5                ;MOV DATA TO IBUS* REGISTER 5
4252  014756                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4253  014756  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4254  014762  121125                            121005!<5*20>           ;READ FROM IBUS* REGISTER 5
4255  014764  116104  000005                    MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
4256  014770  120504                            CMPB    R5,R4           ;DATA CORRECT?
4257  014772  001414                            BEQ     68$             ;BR IF YES
4258  014774                                    BERROR  27              ;ERROR
4259  015014  104455                            TRAP    C$ERDF
```

```
 4260  015016  000033                        .WORD   27
 4261  015020  005525                        .WORD   EM27
 4262  015022  007644                        .WORD   ERR27
 4263  015024                        68$:    ESCAPE  SEG
 4264  015024  104410                        TRAP    C$ESCAPE
 4265  015026  000012                        .WORD   10001$-.
 4266  015030  005105                        COM     R5              ;CHANGE TO FLOATING 1
 4267  015032  000241                        CLC                     ;CLEAR CARRY
 4268  015034  106105                        ROLB    R5              ;SHIFT BIT IN R5
 4269  015036  001341                        BNE     67$             ;IF R5=0 THEN DONE
 4270  015040                                ENDSEG
 4271  015040                        10001$:
 4272  015040  104405                        TRAP    C$ESEG
 4273  015042                        ENDTST
 4274  015042                        L10072:
 4275  015042  104401                        TRAP    C$ETST
 4276
 4277  015044                                BADHEAD
 4278                                        ;******************** TEST 18 ********************
 4279                                        ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
 4280                                        ;*FLOAT A 1 THOUGH IBUS* REGISTER 10
 4281                                        ;*FLOAT A 0 THROUGH IBUS* REGISTER 10
 4282  015044                                BADHEAD
 4283                                        ;******************** TEST 18 ********************
 4284
 4285  015044                        BGNTST
 4286  015044                        T18::
 4287  015044                                MSTCLR                  ;MASTER CLEAR M8200,4,7
 4288  015044  004537  003156                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 4289  015050  012737  000010  002624        MOV     #10,MR0         ;SAVE REGISTER ADDRESS FOR TYPEOUT
 4290  015056  012705  000001                MOV     #1,R5           ;START WITH BIT 0
 4291  015062                                MYINT
 4292  015062  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 4293  015066                                BGNSEG
 4294  015066  104404                        TRAP    C$BSEG
 4295  015070                        64$:
 4296  015070  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4297  015074  042761  000141  000004        BIC     #141,4(R1)      ;CLEAR UNWANTED BITS
 4298  015102                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4299  015102  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 4300  015106  121110                        121100!10               ;MOV DATA TO IBUS* REGISTER 10
 4301  015110                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4302  015110  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 4303  015114  121205                        121005!<10*20>          ;READ FROM IBUS* REGISTER 10
 4304  015116  010502                        MOV     R5,R2
 4305  015120  042705  000141                BIC     #141,R5         ;CLEAR UNWANTED BITS
 4306  015124  116104  000005                MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 4307  015130  042704  000140                BIC     #140,R4         ;CLEAR UNWANTED BITS
 4308  015134  120504                        CMPB    R5,R4           ;DATA CORRECT?
 4309  015136  001414                        BEQ     65$             ;BR IF YES
 4310  015140                                BERROR  27              ;ERROR
 4311  015160  104455                        TRAP    C$ERDF
 4312  015162  000033                        .WORD   27
 4313  015164  005525                        .WORD   EM27
 4314  015166  007644                        .WORD   ERR27
 4315  015170                        65$:    ESCAPE  SEG
```

```
4316  015170  104410                          TRAP    C$ESCAPE
4317  015172  000012                          .WORD   10000$-.
4318  015174  010205                          MOV     R2,R5
4319  015176  000241                          CLC                     ;CLEAR CARRY
4320  015200  106105                          ROLB    R5              ;SHIFT BIT IN R5
4321  015202  001332                          BNE     64$             ;IF R5=0 THEN DONE
4322  015204                                  ENDSEG
4323  015204                          10000$:
4324  015204  104405                          TRAP    C$ESEG
4325  015206  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
4326                                  ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
4327  015212                                  BGNSEG
4328  015212  104404                          TRAP    C$BSEG
4329  015214                          67$:
4330  015214  005105                          COM     R5
4331  015216  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
4332  015222  042761  000141  000004         BIC     #141,4(R1)      ;CLEAR UNWANTED BITS
4333  015230                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4334  015230  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4335  015234  121110                          121100!10               ;MOV DATA TO IBUS* REGISTER 10
4336  015236                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4337  015236  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4338  015242  121205                          121005!<10*20>          ;READ FROM IBUS* REGISTER 10
4339  015244  010502                          MOV     R5,R2
4340  015246  042705  000141                  BIC     #141,R5         ;CLEAR UNWANTED BITS
4341  015252  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
4342  015256  042704  000140                  BIC     #140,R4         ;CLEAR UNWANTED BITS
4343  015262  120504                          CMPB    R5,R4           ;DATA CORRECT?
4344  015264  001414                          BEQ     68$             ;BR IF YES
4345  015266                                  BERROR  27              ;ERROR
4346  015306  104455                          TRAP    C$ERDF
4347  015310  000033                          .WORD   27
4348  015312  005525                          .WORD   EM27
4349  015314  007644                          .WORD   ERR27
4350  015316                          68$:    ESCAPE  SEG
4351  015316  104410                          TRAP    C$ESCAPE
4352  015320  000014                          .WORD   10001$-.
4353  015322  010205                          MOV     R2,R5
4354  015324  005105                          COM     R5              ;CHANGE TO FLOATING 1
4355  015326  000241                          CLC                     ;CLEAR CARRY
4356  015330  106105                          ROLB    R5              ;SHIFT BIT IN R5
4357  015332  001330                          BNE     67$             ;IF R5=0 THEN DONE
4358  015334                                  ENDSEG
4359  015334                          10001$:
4360  015334  104405                          TRAP    C$ESEG
4361  015336                          ENDTST
4362  015336                          L10073:
4363  015336  104401                          TRAP    C$ETST
4364
4365  015340                                  BADHEAD
4366                                          ;******************** TEST 19 ********************
4367                                          ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
4368                                          ;*FLOAT A 1 THOUGH IBUS* REGISTER 11
4369                                          ;*FLOAT A 0 THROUGH IBUS* REGISTER 11
4370  015340                                  BADHEAD
4371                                          ;******************** TEST 19 ********************
```

```
 4372
 4373  015340                          BGNTST
 4374  015340                          T19::
 4375  015340                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 4376  015340  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 4377  015344  012737  000011  002624          MOV     #11,MR0         ;SAVE REGISTER ADDRESS FOR TYPEOUT
 4378  015352  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
 4379  015356                                  MYINT
 4380  015356  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 4381  015362                                  BGNSEG
 4382  015362  104404                          TRAP    C$BSEG
 4383  015364                          64$:
 4384  015364  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4385  015370  042761  000262  000004          BIC     #262,4(R1)      ;CLEAR UNWANTED BITS
 4386  015376                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4387  015376  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 4388  015402  121111                          121100!11               ;MOV DATA TO IBUS* REGISTER 11
 4389  015404                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4390  015404  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 4391  015410  121225                          121005!<11*20>          ;READ FROM IBUS* REGISTER 11
 4392  015412  010502                          MOV     R5,R2
 4393  015414  042705  000262                  BIC     #262,R5         ;CLEAR UNWANTED BITS
 4394  015420  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 4395  015424  042704  000020                  BIC     #20,R4
 4396  015430  120504                          CMPB    R5,R4           ;DATA CORRECT?
 4397  015432  001414                          BEQ     65$             ;BR IF YES
 4398  015434                                  BERROR  27              ;ERROR
 4399  015454  104455                          TRAP    C$ERDF
 4400  015456  000033                          .WORD   27
 4401  015460  005525                          .WORD   EM27
 4402  015462  007644                          .WORD   ERR27
```

```
 4403  015464                          65$:    ESCAPE  SEG
 4404  015464  104410                          TRAP    C$ESCAPE
 4405  015466  000012                          .WORD   10000$-.
 4406  015470  010205                          MOV     R2,R5
 4407  015472  000241                          CLC                     ;CLEAR CARRY
 4408  015474  106105                          ROLB    R5              ;SHIFT BIT IN R5
 4409  015476  001332                          BNE     64$             ;IF R5=0 THEN DONE
 4410  015500                                  ENDSEG
 4411  015500                          10000$:
 4412  015500  104405                          TRAP    C$ESEG
 4413  015502  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
 4414                                  ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
 4415  015506                                  BGNSEG
 4416  015506  104404                          TRAP    C$BSEG
 4417  015510                          67$:
 4418  015510  005105                          COM     R5
 4419  015512  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4420  015516  042761  000262  000004         BIC     #262,4(R1)      ;CLEAR UNWANTED BITS
 4421  015524                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4422  015524  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 4423  015530  121111                          121100!11               ;MOV DATA TO IBUS* REGISTER 11
 4424  015532                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4425  015532  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 4426  015536  121225                          121005!<11*20>          ;READ FROM IBUS* REGISTER 11
```

```
4427  015540  010502                        MOV     R5,R2
4428  015542  042705  000262                BIC     #262,R5         ;CLEAR UNWANTED BITS
4429  015546  052705  000020                BIS     #20,R5          ;ADD THESE BITS
4430  015552  116104  000005                MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
4431  015556  120504                        CMPB    R5,R4           ;DATA CORRECT?
4432  015560  001414                        BEQ     68$             ;BR IF YES
4433  015562                                BERROR  27              ;ERROR
4434  015602  104455                        TRAP    C$ERDF
4435  015604  000033                        .WORD   27
4436  015606  005525                        .WORD   EM27
4437  015610  007644                        .WORD   ERR27
4438  015612                        68$:    ESCAPE  SEG
4439  015612  104410                        TRAP    C$ESCAPE
4440  015614  000014                        .WORD   10001$-.
4441  015616  010205                        MOV     R2,R5
4442  015620  005105                        COM     R5              ;CHANGE TO FLOATING 1
4443  015622  000241                        CLC                     ;CLEAR CARRY
4444  015624  106105                        ROLB    R5              ;SHIFT BIT IN R5
4445  015626  001330                        BNE     67$             ;IF R5=0 THEN DONE
4446  015630                                ENDSEG
4447  015630                        10001$:
4448  015630  104405                        TRAP    C$ESEG
4449  015632                        ENDTST
4450  015632                        L10074:
4451  015632  104401                        TRAP    C$ETST
4452
4453  015634                                BADHEAD
4454                                        ;******************** TEST 20 *********************
4455                                        ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
4456                                        ;*FLOAT A 1 THOUGH IBUS REGISTER 0
4457                                        ;*FLOAT A 0 THROUGH IBUS REGISTER 0
4458  015634                                BADHEAD
4459                                        ;******************** TEST 20 *********************
4460
4461  015634                        BGNTST
4462  015634                        T20::
4463  015634                                MSTCLR                  ;MASTER CLEAR M8200,4,7
4464  015634  004537  003156                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
4465  015640  012737  000000  002624        MOV     #0,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
4466  015646  012705  000001                MOV     #1,R5           ;START WITH BIT 0
4467  015652                                MYINT
4468  015652  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
4469  015656                                BGNSEG
4470  015656  104404                        TRAP    C$BSEG
4471  015660                        64$:
4472  015660  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
4473  015664                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4474  015664  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
4475  015670  122100                        122100                  ;MOV DATA TO IBUS* REGISTER 0
4476  015672                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4477  015672  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
4478  015676  021005                        21005                   ;READ FROM IBUS* REGISTER 0
4479  015700  116104  000005                MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
4480  015704  120504                        CMPB    R5,R4           ;DATA CORRECT?
4481  015706  001414                        BEQ     65$             ;BR IF YES
4482  015710                                BERROR  29              ;ERROR
```

```
 4483  015730  104455                        TRAP    C$ERDF
 4484  015732  000035                        .WORD   29
 4485  015734  005605                        .WORD   EM29
 4486  015736  010004                        .WORD   ERR29
 4487  015740                        65$:    ESCAPE  SEG
 4488  015740  104410                        TRAP    C$ESCAPE
 4489  015742  000010                        .WORD   10000$-.
 4490  015744  000241                        CLC                     ;CLEAR CARRY
 4491  015746  106105                        ROLB    R5              ;SHIFT BIT IN R5
 4492  015750  001343                        BNE     64$             ;IF R5=0 THEN DONE
 4493  015752                                ENDSEG
 4494  015752                        10000$:
 4495  015752  104405                        TRAP    C$ESEG
 4496  015754  012705  000001                MOV     #1,R5           ;START WITH BIT 0
 4497                                ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
 4498  015760                                BGNSEG
 4499  015760  104404                        TRAP    C$BSEG
 4500  015762                        67$:
 4501  015762  005105                        COM     R5
 4502  015764  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4503  015770                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4504  015770  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 4505  015774  122100                        122100                  ;MOV DATA TO IBUS* REGISTER 0
 4506  015776                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4507  015776  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 4508  016002  021005                        21005                   ;READ FROM IBUS* REGISTER 0
 4509  016004  116104  000005                MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 4510  016010  120504                        CMPB    R5,R4           ;DATA CORRECT?
 4511  016012  001414                        BEQ     68$             ;BR IF YES
 4512  016014                                BERROR  29              ;ERROR
 4513  016034  104455                        TRAP    C$ERDF
 4514  016036  000035                        .WORD   29
 4515  016040  005605                        .WORD   EM29
 4516  016042  010004                        .WORD   ERR29
 4517  016044                        68$:    ESCAPE  SEG
 4518  016044  104410                        TRAP    C$ESCAPE
 4519  016046  000012                        .WORD   10001$-.
 4520  016050  005105                        COM     R5              ;CHANGE TO FLOATING 1
 4521  016052  000241                        CLC                     ;CLEAR CARRY
 4522  016054  106105                        ROLB    R5              ;SHIFT BIT IN R5
 4523  016056  001341                        BNE     67$             ;IF R5=0 THEN DONE
 4524  016060                                ENDSEG
 4525  016060                        10001$:
 4526  016060  104405                        TRAP    C$ESEG
 4527  016062                        ENDTST
 4528  016062                        L10075:
 4529  016062  104401                        TRAP    C$ETST
 4530
 4531  016064                                BADHEAD
 4532                                ;******************** TEST 21 ********************
 4533                                ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
 4534                                ;*FLOAT A 1 THOUGH IBUS REGISTER 1
 4535                                ;*FLOAT A 0 THROUGH IBUS REGISTER 1
 4536  016064                                BADHEAD
 4537                                ;******************** TEST 21 ********************
 4538
```

```
 4539  016064                                BGNTST
 4540  016064                                T21::
 4541  016064                                      MSTCLR                  ;MASTER CLEAR M8200,4,7
 4542  016064  004537  003156                      JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 4543  016070  012737  000001  002624              MOV     #1,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
 4544  016076  012705  000001                      MOV     #1,R5           ;START WITH BIT 0
 4545  016102                                      MYINT
 4546  016102  013701  002716                      MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 4547  016106                                      BGNSEG
 4548  016106  104404                              TRAP    C$BSEG
 4549  016110                                64$:
 4550  016110  010561  000004                      MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4551  016114                                      ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4552  016114  004537  003244                      JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 4553  016120  122101                              122100!1                ;MOV DATA TO IBUS* REGISTER 1
 4554  016122                                      ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4555  016122  004537  003244                      JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 4556  016126  021025                              21005!<1*20>            ;READ FROM IBUS* REGISTER 1
 4557  016130  116104  000005                      MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
 4558  016134  120504                              CMPB    R5,R4           ;DATA CORRECT?
 4559  016136  001414                              BEQ     65$             ;BR IF YES
 4560  016140                                      BERROR  29              ;ERROR
 4561  016160  104455                              TRAP    C$ERDF
 4562  016162  000035                              .WORD   29
 4563  016164  005605                              .WORD   EM29
 4564  016166  010004                              .WORD   ERR29
 4565  016170                                65$:   ESCAPE  SEG
 4566  016170  104410                              TRAP    C$ESCAPE
 4567  016172  000010                              .WORD   10000$-.
 4568  016174  000241                              CLC                     ;CLEAR CARRY
 4569  016176  106105                              ROLB    R5              ;SHIFT BIT IN R5
 4570  016200  001343                              BNE     64$             ;IF R5=0 THEN DONE
 4571  016202                                      ENDSEG
 4572  016202                                10000$:
 4573  016202  104405                              TRAP    C$ESEG
 4574  016204  012705  000001                      MOV     #1,R5           ;START WITH BIT 0
 4575                                  ;69$:       COM     R5              ;CHANGE TO FLOATING ZERO
 4576  016210                                      BGNSEG
 4577  016210  104404                              TRAP    C$BSEG
 4578  016212                                67$:
 4579  016212  005105                              COM     R5
 4580  016214  010561  000004                      MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4581  016220                                      ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4582  016220  004537  003244                      JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 4583  016224  122101                              122100!1                ;MOV DATA TO IBUS* REGISTER 1
 4584  016226                                      ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4585  016226  004537  003244                      JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 4586  016232  021025                              21005!<1*20>            ;READ FROM IBUS* REGISTER 1
 4587  016234  116104  000005                      MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
 4588  016240  120504                              CMPB    R5,R4           ;DATA CORRECT?
 4589  016242  001414                              BEQ     68$             ;BR IF YES
 4590  016244                                      BERROR  29              ;ERROR
 4591  016264  104455                              TRAP    C$ERDF
 4592  016266  000035                              .WORD   29
 4593  016270  005605                              .WORD   EM29
 4594  016272  010004                              .WORD   ERR29
```

```
 4595  016274                          68$:    ESCAPE  SEG
 4596  016274  104410                          TRAP    C$ESCAPE
 4597  016276  000012                          .WORD   10001$-.
 4598  016300  005105                          COM     R5              ;CHANGE TO FLOATING 1
 4599  016302  000241                          CLC                     ;CLEAR CARRY
 4600  016304  106105                          ROLB    R5              ;SHIFT BIT IN R5
 4601  016306  001341                          BNE     67$             ;IF R5=0 THEN DONE
 4602  016310                                  ENDSEG
 4603  016310                          10001$:
 4604  016310  104405                          TRAP    C$ESEG
 4605  016312                          ENDTST
 4606  016312                          L10076:
 4607  016312  104401                          TRAP    C$ETST
 4608
 4609  016314                                  BADHEAD
 4610                                          ;******************** TEST 22 ********************
 4611                                          ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
 4612                                          ;*FLOAT A 1 THOUGH IBUS REGISTER 2
 4613                                          ;*FLOAT A 0 THROUGH IBUS REGISTER 2
 4614  016314                                  BADHEAD
 4615                                          ;******************** TEST 22 ********************
 4616
 4617  016314                          BGNTST
 4618  016314                          T22::
 4619  016314                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 4620  016314  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 4621  016320  012737  000002  002624          MOV     #2,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
 4622  016326  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
 4623  016332                                  MYINT
 4624  016332  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 4625  016336                                  BGNSE ;
 4626  016336  104404                          TRAP    C$BSEG
 4627  016340                          64$:
 4628  016340  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4629  016344                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4630  016344  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 4631  016350  122102                          122100!2                ;MOV DATA TO IBUS* REGISTER 2
 4632  016352                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4633  016352  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 4634  016356  021045                          21005!<2*20>            ;READ FROM IBUS* REGISTER 2
 4635  016360  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
 4636  016364  120504                          CMPB    R5,R4           ;DATA CORRECT?
 4637  016366  001414                          BEQ     65$             ;BR IF YES
 4638  016370                                  BERROR  29              ;ERROR
 4639  016410  104455                          TRAP    C$ERDF
 4640  016412  000035                          .WORD   29
 4641  016414  005605                          .WORD   EM29
 4642  016416  010004                          .WORD   ERR29
 4643  016420                          65$:    ESCAPE  SEG
 4644  016420  104410                          TRAP    C$ESCAPE
 4645  016422  000010                          .WORD   10000$-.
 4646  016424  000241                          CLC                     ;CLEAR CARRY
 4647  016426  106105                          ROLB    R5              ;SHIFT BIT IN R5
 4648  016430  001343                          BNE     64$             ;IF R5=0 THEN DONE
 4649  016432                                  ENDSEG
 4650  016432                          10000$:
```

```
 4651  016432  104405                          TRAP    C$ESEG
 4652  016434  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
 4653                                  ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
 4654  016440                                  BGNSEG
 4655  016440  104404                          TRAP    C$BSEG
 4656  016442                                67$:
 4657  016442  005105                          COM     R5
 4658  016444  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4659  016450                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4660  016450  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 4661  016454  122102                          122100!2                ;MOV DATA TO IBUS* REGISTER 2
 4662  016456                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4663  016456  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 4664  016462  021045                          21005!<2*20>            ;READ FROM IBUS* REGISTER 2
 4665  016464  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 4666  016470  120504                          CMPB    R5,R4           ;DATA CORRECT?
 4667  016472  001414                          BEQ     68$             ;BR IF YES
 4668  016474                                  BERROR  29              ;ERROR
 4669  016514  104455                          TRAP    C$ERDF
 4670  016516  000035                          .WORD   29
 4671  016520  005605                          .WORD   EM29
 4672  016522  010004                          .WORD   ERR29
 4673  016524                                68$:    ESCAPE  SEG
 4674  016524  104410                          TRAP    C$ESCAPE
 4675  016526  000012                          .WORD   10001$-.
 4676  016530  005105                          COM     R5              ;CHANGE TO FLOATING 1
 4677  016532  000241                          CLC                     ;CLEAR CARRY
 4678  016534  106105                          ROLB    R5              ;SHIFT BIT IN R5
 4679  016536  001341                          BNE     67$             ;IF R5=0 THEN DONE
 4680  016540                                  ENDSEG
 4681  016540                                10001$:
 4682  016540  104405                          TRAP    C$ESEG
 4683  016542                                ENDTST
 4684  016542                                L10077:
 4685  016542  104401                          TRAP    C$ETST
 4686
 4687  016544                                  BADHEAD
 4688                                          ;******************** TEST 23 ********************
 4689                                          ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
 4690                                          ;*FLOAT A 1 THOUGH IBUS REGISTER 3
 4691                                          ;*FLOAT A 0 THROUGH IBUS REGISTER 3
 4692  016544                                  BADHEAD
 4693                                          ;******************** TEST 23 ********************
 4694
 4695  016544                                BGNTST
 4696  016544                                T23::
 4697  016544                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 4698  016544  004537  003156                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 4699  016550  012737  000003  002624          MOV     #3,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
 4700  016556  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
 4701  016562                                  MYINT
 4702  016562  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 4703  016566                                  BGNSEG
 4704  016566  104404                          TRAP    C$BSEG
 4705  016570                                64$:
 4706  016570  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
```

```
4707  016574                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4708  016574  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4709  016600  122103                            122100!3                ;MOV DATA TO IBUS* REGISTER 3
4710  016602                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4711  016602  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4712  016606  021065                            21005!<3*20>            ;READ FROM IBUS* REGISTER 3
4713  016610  116104  000005                    MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
4714  016614  120504                            CMPB    R5,R4           ;DATA CORRECT?
4715  016616  001414                            BEQ     65$             ;BR IF YES
4716  016620                                    BERROR  29              ;ERROR
4717  016640  104455                            TRAP    C$ERDF
4718  016642  000035                            .WORD   29
4719  016644  005605                            .WORD   EM29
4720  016646  010004                            .WORD   ERR29
4721  016650                            65$:    ESCAPE  SEG
4722  016650  104410                            TRAP    C$ESCAPE
4723  016652  000010                            .WORD   10000$-.
4724  016654  000241                            CLC                     ;CLEAR CARRY
4725  016656  106105                            ROLB    R5              ;SHIFT BIT IN R5
4726  016660  001343                            BNE     64$             ;IF R5=0 THEN DONE
4727  016662                                    ENDSEG
4728  016662                            10000$:
4729  016662  104405                            TRAP    C$ESEG
4730  016664  012705  000001                    MOV     #1,R5           ;START WITH BIT 0
4731                                    ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
4732  016670                                    BGNSEG
4733  016670  104404                            TRAP    C$BSEG
4734  016672                            67$:
4735  016672  005105                            COM     R5
4736  016674  010561  000004                    MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
4737  016700                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4738  016700  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4739  016704  122103                            122100!3                ;MOV DATA TO IBUS* REGISTER 3
4740  016706                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4741  016706  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4742  016712  021065                            21005!<3*20>            ;READ FROM IBUS* REGISTER 3
4743  016714  116104  000005                    MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
4744  016720  120504                            CMPB    R5,R4           ;DATA CORRECT?
4745  016722  001414                            BEQ     68$             ;BR IF YES
4746  016724                                    BERROR  29              ;ERROR
4747  016744  104455                            TRAP    C$ERDF
4748  016746  000035                            .WORD   29
4749  016750  005605                            .WORD   EM29
4750  016752  010004                            .WORD   ERR29
4751  016754                            68$:    ESCAPE  SEG
4752  016754  104410                            TRAP    C$ESCAPE
4753  016756  000012                            .WORD   10001$-.
4754  016760  005105                            COM     R5              ;CHANGE TO FLOATING 1
4755  016762  000241                            CLC                     ;CLEAR CARRY
4756  016764  106105                            ROLB    R5              ;SHIFT BIT IN R5
4757  016766  001341                            BNE     67$             ;IF R5=0 THEN DONE
4758  016770                                    ENDSEG
4759  016770                            10001$:
4760  016770  104405                            TRAP    C$ESEG
4761  016772                            ENDTST
4762  016772                            L10100:
```

```
 4763  016772  104401                        TRAP    C$ETST
 4764
 4765  016774                                BADHEAD
 4766                                        ;******************** TEST 24 ********************
 4767                                        ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
 4768                                        ;*FLOAT A 1 THOUGH IBUS REGISTER 4
 4769                                        ;*FLOAT A 0 THROUGH IBUS REGISTER 4
 4770  016774                                BADHEAD
 4771                                        ;******************** TEST 24 ********************
 4772
 4773  016774                        BGNTST
 4774  016774                        T24::
 4775  016774                                MSTCLR                  ;MASTER CLEAR M8200,4,7
 4776  016774  004537  003156                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 4777  017000  012737  000004  002624        MOV     #4,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
 4778  017006  012705  000001                MOV     #1,R5           ;START WITH BIT 0
 4779  017012                                MYINT
 4780  017012  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 4781  017016                                BGNSEG
 4782  017016  104404                        TRAP    C$BSEG
 4783  017020                        64$:
 4784  017020  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4785  017024                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4786  017024  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 4787  017030  122104                        122100!4                ;MOV DATA TO IBUS* REGISTER 4
 4788  017032                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4789  017032  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 4790  017036  021105                        21005!<4*20>            ;READ FROM IBUS* REGISTER 4
 4791  017040  116104  000005                MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 4792  017044  120504                        CMPB    R5,R4           ;DATA CORRECT?
 4793  017046  001414                        BEQ     65$             ;BR IF YES
 4794  017050                                BERROR  29              ;ERROR
 4795  017070  104455                        TRAP    C$ERDF
 4796  017072  000035                        .WORD   29
 4797  017074  005605                        .WORD   EM29
 4798  017076  010004                        .WORD   ERR29
 4799  017100                        65$:    ESCAPE  SEG
 4800  017100  104410                        TRAP    C$ESCAPE
 4801  017102  000010                        .WORD   10000$-.
 4802  017104  000241                        CLC                     ;CLEAR CARRY
 4803  017106  106105                        ROLB    R5              ;SHIFT BIT IN R5
 4804  017110  001343                        BNE     64$             ;IF R5=0 THEN DONE
 4805  017112                                ENDSEG
 4806  017112                        10000$:
 4807  017112  104405                        TRAP    C$ESEG
 4808  017114  012705  000001                MOV     #1,R5           ;START WITH BIT 0
 4809                                ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
 4810  017120                                BGNSEG
 4811  017120  104404                        TRAP    C$BSEG
 4812  017122                        67$:
 4813  017122  005105                        COM     R5
 4814  017124  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4815  017130                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4816  017130  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 4817  017134  122104                        122100!4                ;MOV DATA TO IBUS* REGISTER 4
 4818  017136                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
```

```
4819  017136  004537  003244          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
4820  017142  021105                  21005!<4*20>            ;READ FROM IBUS* REGISTER 4
4821  017144  116104  000005          MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
4822  017150  120504                  CMPB    R5,R4           ;DATA CORRECT?
4823  017152  001414                  BEQ     68$             ;BR IF YES
4824  017154                          BERROR  29              ;ERROR
4825  017174  104455                  TRAP    C$ERDF
4826  017176  000035                  .WORD   29
4827  017200  005605                  .WORD   EM29
4828  017202  010004                  .WORD   ERR29
4829  017204                  68$:    ESCAPE  SEG
4830  017204  104410                  TRAP    C$ESCAPE
4831  017206  000012                  .WORD   10001$-.
4832  017210  005105                  COM     R5              ;CHANGE TO FLOATING 1
4833  017212  000241                  CLC                     ;CLEAR CARRY
4834  017214  106105                  ROLB    R5              ;SHIFT BIT IN R5
4835  017216  001341                  BNE     67$             ;IF R5=0 THEN DONE
4836  017220                          ENDSEG
4837  017220                  10001$:
4838  017220  104405                  TRAP    C$ESEG
4839  017222                  ENDTST
4840  017222                  L10101:
4841  017222  104401                  TRAP    C$ETST
4842
4843  017224                          BADHEAD
4844                                  ;******************** TEST 25 ********************
4845                                  ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
4846                                  ;*FLOAT A 1 THOUGH IBUS REGISTER 5
4847                                  ;*FLOAT A 0 THROUGH IBUS REGISTER 5
4848  017224                          BADHEAD
4849                                  ;******************** TEST 25 ********************
4850
4851  017224                  BGNTST
4852  017224                  T25::
4853  017224                          MSTCLR                  ;MASTER CLEAR M8200,4,7
4854  017224  004537  003156          JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
4855  017230  012737  000005  002624  MOV     #5,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
4856  017236  012705  000001          MOV     #1,R5           ;START WITH BIT 0
4857  017242                          MYINT
4858  017242  013701  002716          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
4859  017246                          BGNSEG
4860  017246  104404                  TRAP    C$BSEG
4861  017250                  64$:
4862  017250  010561  000004          MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
4863  017254                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4864  017254  004537  003244          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
4865  017260  122105                  122100!5                ;MOV DATA TO IBUS* REGISTER 5
4866  017262                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
4867  017262  004537  003244          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
4868  017266  021105                  21005!<5*20>            ;READ FROM IBUS* REGISTER 5
4869  017270  116104  000005          MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
4870  017274  120504                  CMPB    R5,R4           ;DATA CORRECT?
4871  017276  001414                  BEQ     65$             ;BR IF YES
4872  017300                          BERROR  29              ;ERROR
4873  017320  104455                  TRAP    C$ERDF
4874  017322  000035                  .WORD   29
```

```
 4875  017324  005605                          .WORD   EM29
 4876  017326  010004                          .WORD   ERR29
 4877  017330                          65$:    ESCAPE  SEG
 4878  017330  104410                          TRAP    C$ESCAPE
 4879  017332  000010                          .WORD   10000$-.
 4880  017334  000241                          CLC                     ;CLEAR CARRY
 4881  017336  106105                          ROLB    R5              ;SHIFT BIT IN R5
 4882  017340  001343                          BNE     64$             ;IF R5=0 THEN DONE
 4883  017342                                  ENDSEG
 4884  017342                          10000$:
 4885  017342  104405                          TRAP    C$ESEG
 4886  017344  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
 4887                                  ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
 4888  017350                                  BGNSEG
 4889  017350  104404                          TRAP    C$BSEG
 4890  017352                          67$:
 4891  017352  005105                          COM     R5
 4892  017354  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4893  017360                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4894  017360  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 4895  017364  122105                          122100!5                ;MOV DATA TO IBUS* REGISTER 5
 4896  017366                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4897  017366  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 4898  017372  021125                          21005!<5*20>            ;READ FROM IBUS* REGISTER 5
 4899  017374  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 4900  017400  120504                          CMPB    R5,R4           ;DATA CORRECT?
 4901  017402  001414                          BEQ     68$             ;BR IF YES
 4902  017404                                  BERROR  29              ;ERROR
 4903  017424  104455                          TRAP    C$ERDF
 4904  017426  000035                          .WORD   29
 4905  017430  005605                          .WORD   EM29
 4906  017432  010004                          .WORD   ERR29
 4907  017434                          68$:    ESCAPE  SEG
 4908  017434  104410                          TRAP    C$ESCAPE
 4909  017436  000012                          .WORD   10001$-.
 4910  017440  005105                          COM     R5              ;CHANGE TO FLOATING 1
 4911  017442  000241                          CLC                     ;CLEAR CARRY
 4912  017444  106105                          ROLB    R5              ;SHIFT BIT IN R5
 4913  017446  001341                          BNE     67$             ;IF R5=0 THEN DONE
 4914  017450                                  ENDSEG
 4915  017450                          10001$:
 4916  017450  104405                          TRAP    C$ESEG
 4917  017452                          ENDTST
 4918  017452                          L10102:
 4919  017452  104401                          TRAP    C$ETST
 4920
 4921  017454                                  BADHEAD
 4922                                          ;******************** TEST 26 ********************
 4923                                          ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
 4924                                          ;*FLOAT A 1 THOUGH IBUS REGISTER 6
 4925                                          ;*FLOAT A 0 THROUGH IBUS REGISTER 6
 4926  017454                                  BADHEAD
 4927                                          ;******************** TEST 26 ********************
 4928
 4929  017454                          BGNTST
 4930  017454                          T26::
```

```
 4931  017454                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 4932  017454  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 4933  017460  012737  000006  002624          MOV     #6,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
 4934  017466  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
 4935  017472                                  MYINT
 4936  017472  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 4937  017476                                  BGNSEG
 4938  017476  104404                          TRAP    C$BSEG
 4939  017500                          64$:
 4940  017500  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4941  017504                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4942  017504  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 4943  017510  122106                          122100!6                ;MOV DATA TO IBUS* REGISTER 6
 4944  017512                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4945  017512  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 4946  017516  021145                          21005!<6*20>            ;READ FROM IBUS* REGISTER 6
 4947  017520  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 4948  017524  120504                          CMPB    R5,R4           ;DATA CORRECT?
 4949  017526  001414                          BEQ     65$             ;BR IF YES
 4950  017530                                  BERROR  29              ;ERROR
 4951  017550  104455                          TRAP    C$ERDF
 4952  017552  000035                          .WORD   29
 4953  017554  005605                          .WORD   EM29
 4954  017556  010004                          .WORD   ERR29
 4955  017560                          65$:    ESCAPE  SEG
 4956  017560  104410                          TRAP    C$ESCAPE
 4957  017562  000010                          .WORD   10000$-.
 4958  017564  000241                          CLC                     ;CLEAR CARRY
 4959  017566  106105                          ROLB    R5              ;SHIFT BIT IN R5
 4960  017570  001343                          BNE     64$             ;IF R5=0 THEN DONE
 4961  017572                                  ENDSEG
 4962  017572                          10000$:
 4963  017572  104405                          TRAP    C$ESEG
 4964  017574  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
 4965                                  ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
 4966  017600                                  BGNSEG
 4967  017600  104404                          TRAP    C$BSEG
 4968  017602                          67$:
 4969  017602  005105                          COM     R5
 4970  017604  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4971  017610                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4972  017610  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 4973  017614  122106                          122100!6                ;MOV DATA TO IBUS* REGISTER 6
 4974  017616                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4975  017616  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 4976  017622  021145                          21005!<6*20>            ;READ FROM IBUS* REGISTER 6
 4977  017624  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 4978  017630  120504                          CMPB    R5,R4           ;DATA CORRECT?
 4979  017632  001414                          BEQ     68$             ;BR IF YES
 4980  017634                                  BERROR  29              ;ERROR
 4981  017654  104455                          TRAP    C$ERDF
 4982  017656  000035                          .WORD   29
 4983  017660  005605                          .WORD   EM29
 4984  017662  010004                          .WORD   ERR29
 4985  017664                          68$:    ESCAPE  SEG
 4986  017664  104410                          TRAP    C$ESCAPE
```

```
 4987  017666  000012                        .WORD   10001$-.
 4988  017670  005105                        COM     R5              ;CHANGE TO FLOATING 1
 4989  017672  000241                        CLC                     ;CLEAR CARRY
 4990  017674  106105                        ROLB    R5              ;SHIFT BIT IN R5
 4991  017676  001341                        BNE     67$             ;IF R5=0 THEN DONE
 4992  017700                                ENDSEG
 4993  017700                        10001$:
 4994  017700  104405                        TRAP    C$ESEG
 4995  017702                        ENDTST
 4996  017702                        L10103:
 4997  017702  104401                        TRAP    C$ETST
 4998
 4999  017704                                BADHEAD
 5000                                        ;******************** TEST 27 ********************
 5001                                        ;*MICRO PROCEOR IBUS* REGISTER WRITE/READ TEST
 5002                                        ;*FLOAT A 1 THOUGH IBUS* REGISTER 7
 5003                                        ;*FLOAT A 0 THROUGH IBUS* REGISTER 7
 5004  017704                                BADHEAD
 5005                                        ;******************** TEST 27 ********************
 5006
 5007  017704                        BGNTST
 5008  017704                        T27::
 5009  017704                                MSTCLR                  ;MASTER CLEAR M8200,4,7
 5010  017704  004537  003156                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 5011  017710  012737  000007  002624        MOV     #7,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
 5012  017716  012705  000001                MOV     #1,R5           ;START WITH BIT 0
 5013  017722                                MYINT
 5014  017722  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 5015  017726                                BGNSEG
 5016  017726  104404                        TRAP    C$BSEG
 5017  017730                        64$:
 5018  017730  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 5019  017734                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 5020  017734  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 5021  017740  122107                        122100!7                ;MOV DATA TO IBUS* REGISTER 7
 5022  017742                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 5023  017742  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 5024  017746  021165                        21005!<7*20>            ;READ FROM IBUS* REGISTER 7
 5025  017750  116104  000005                MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 5026  017754  120504                        CMPB -  R5,R4           ;DATA CORRECT?
 5027  017756  001414                        BEQ     65$             ;BR IF YES
 5028  017760                                BERROR  29              ;ERROR
 5029  020000  104455                        TRAP    C$ERDF
 5030  020002  000035                        .WORD   29
 5031  020004  005605                        .WORD   EM29
 5032  020006  010004                        .WORD   ERR29
 5033  020010                        65$:    ESCAPE  SEG
 5034  020010  104410                        TRAP    C$ESCAPE
 5035  020012  000010                        .WORD   10000$-.
 5036  020014  000241                        CLC                     ;CLEAR CARRY
 5037  020016  106105                        ROLB    R5              ;SHIFT BIT IN R5
 5038  020020  001343                        BNE     64$             ;IF R5=0 THEN DONE
 5039  020022                                ENDSEG
 5040  020022                        10000$:
 5041  020022  104405                        TRAP    C$ESEG
 5042  020024  012705  000001                MOV     #1,R5           ;START WITH BIT 0
```

```
5043                                    ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
5044  020030                                    BGNSEG
5045  020030  104404                            TRAP    C$BSEG
5046  020032                            67$:
5047  020032  005105                            COM     R5
5048  020034  010561  000004                    MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5049  020040                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5050  020040  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5051  020044  122107                            122100!7                ;MOV DATA TO IBUS* REGISTER 7
5052  020046                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5053  020046  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5054  020052  021165                            21005!<7*20>            ;READ FROM IBUS* REGISTER 7
5055  020054  116104  000005                    MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
5056  020060  120504                            CMPB    R5,R4           ;DATA CORRECT?
5057  020062  001414                            BEQ     68$             ;BR IF YES
5058  020064                                    BERROR  29              ;ERROR
5059  020104  104455                            TRAP    C$ERDF
5060  020106  000035                            .WORD   29
5061  020110  005605                            .WORD   EM29
5062  020112  010004                            .WORD   ERR29
5063  020114                            68$:    ESCAPE  SEG
5064  020114  104410                            TRAP    C$ESCAPE
5065  020116  000012                            .WORD   10001$-.
5066  020120  005105                            COM     R5              ;CHANGE TO FLOATING 1
5067  020122  000241                            CLC                     ;CLEAR CARRY
5068  020124  106105                            ROLB    R5              ;SHIFT BIT IN R5
5069  020126  001341                            BNE     67$             ;IF R5=0 THEN DONE
5070  020130                                    ENDSEG
5071  020130                            10001$:
5072  020130  104405                            TRAP    C$ESEG
5073  020132                            ENDTST
5074  020132                            L10104:
5075  020132  104401                            TRAP    C$ETST
5076
5077  020134                                    BADHEAD
5078                                            ;******************** TEST 28 ********************
5079                                            ;*MICRO PROCESSOR IBUS DUAL ADDRESS TEST
5080                                            ;*WRITE ALL IBUS REGISTERS WITH INCREMENTING PATTERN
5081                                            ;*READ ALL IBUS REGISTERS TO VERIFY CORRECT ADDRESSING
5082  020134                                    BADHEAD
5083                                            ;******************** TEST 28 ********************
5084
5085  020134                            BGNTST
5086  020134                            T28::
5087  020134                                    MSTCLR                  ;MASTER CLEAR M8200,4,7
5088  020134  004537  003156                    JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
5089  020140  012705  000001                    MOV     #1,R5           ;START WITH A ONE
5090  020144  005002                            CLR     R2              ;R2 CONTAINS ADDRESS OF REGISTER
5091  020146                                    MYINT
5092  020146  013701  002716                    MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5093  020152                                    BGNSEG
5094  020152  104404                            TRAP    C$BSEG
5095  020154  010203                    1$:     MOV     R2,R3           ;R3=REGISTER ADDRESS
5096  020156  010561  000004                    MOV     R5,4(R1)        ;WRITE DATA TO PORT4
5097  020162  042737  000017  020200            BIC     #17,5$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
5098  020170  050337  020200                    BIS     R3,5$           ;ADD ADDRESS TO INSTRUCTION
```

```
5099  020174                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5100  020174  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5101  020200  122100                    5$:     122100                  ;MOVE DATA TO IBUS REGISTER
5102  020202  006303                            ASL     R3              ;SHIFT ADDRESS
5103  020204  006303                            ASL     R3              ;4 TIMES TO GET
5104  020206  006303                            ASL     R3              ;IT TO BITS 4-7
5105  020210  006303                            ASL     R3              ;OF NEXT INSTRUCTION
5106  020212  042737  000360  020230            BIC     #360,6$         ;CLEAR ADDRESS FIELD
5107  020220  050337  020230                    BIS     R3,6$           ;ADD ADDRESS TO INSTRUCTION
5108  020224                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5109  020224  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5110  020230  021005                    6$:     21005                   ;READ FROM IBUS REGISTER
5111  020232  116104  000005                    MOVB    5(R1),R4        ;PUT "FOUND" IN R4
5112  020236  120504                            CMPB    R5,R4           ;IS DATA CORRECT?
5113  020240  001414                            BEQ     2$              ;BR IF YES
5114  020242                                    BERROR  29              ;DATA ERROR
5115  020262  104455                            TRAP    C$ERDF
5116  020264  000035                            .WORD   29
5117  020266  005605                            .WORD   EM29
5118  020270  010004                            .WORD   ERR29
5119  020272                            2$:     ESCAPE  SEG
5120  020272  104410                            TRAP    C$ESCAPE
5121  020274  000014                            .WORD   10000$-.
5122  020276  005205                            INC     R5              ;INCREMENT PATTERN
5123  020300  005202                            INC     R2              ;INCREMENT REGISTER ADDRESS
5124  020302  022702  000010                    CMP     #7+1,R2         ;LAST ADDRESS DONE?
5125  020306  001322                            BNE     1$              ;BR IF NO
5126  020310                                    ENDSEG
5127  020310                            10000$:
5128  020310  104405                            TRAP    C$ESEG
5129  020312  012705  000001                    MOV     #1,R5           ;RESTART PATTERN TO 1
5130  020316  005002                            CLR     R2              ;RESTART AT ADDRESS 0
5131  020320                                    BGNSEG
5132  020320  104404                            TRAP    C$BSEG
5133  020322  005003                            CLR     R3              ;RESTART AT ADDRESS 0
5134  020324  042737  000360  020342    3$:     BIC     #360,7$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
5135  020332  050337  020342                    BIS     R3,7$           ;ADD ADDRESS TO INSTRUCTION
5136  020336                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5137  020336  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5138  020342  021005                    7$:     21005                   ;READ FROM IBUS REGISTER
5139  020344  116104  000005                    MOVB    5(R1),R4        ;PUT "FOUND" IN $GDDAT
5140  020350  120504                            CMPB    R5,R4           ;DATA CORRECT?
5141  020352  001414                            BEQ     4$              ;BR IF YES
5142  020354                                    BERROR  30              ;DUAL ADDRESSING ERROR
5143  020374  104455                            TRAP    C$ERDF
5144  020376  000036                            .WORD   30
5145  020400  004353                            .WORD   EM30
5146  020402  010066                            .WORD   ERR30
5147  020404                            4$:     ESCAPE  SEG
5148  020404  104410                            TRAP    C$ESCAPE
5149  020406  000020                            .WORD   10001$-.
5150  020410  005205                            INC     R5              ;INCREMENT PATTERN
5151  020412  005202                            INC     R2              ;NEXT ADDRESS
5152  020414  062703  000020                    ADD     #20,R3          ;ADD 1 TO ADDRESS IN R3(SHIFTED 4 TIMES)
5153  020420  022702  000010                    CMP     #7+1,R2         ;LAST ADDRESS DONE?
5154  020424  001337                            BNE     3$              ;BR IF NO
```

```
5155  020426                                ENDSEG
5156  020426                          10001$:
5157  020426  104405                        TRAP    C$ESEG
5158  020430                          ENDTST
5159  020430                          L10105:
5160  020430  104401                        TRAP    C$ETST
5161
5162  020432                                BADHEAD
5163                                        ;******************** TEST 29 ********************
5164                                        ;*MICRO PROCESSOR BR REGISTER TEST
5165                                        ;*FLOAT A 1 THOUGH THE BR
5166                                        ;*FLOAT A 0 THOUGH THE BR
5167  020432                                BADHEAD
5168                                        ;******************** TEST 29 ********************
5169
5170  020432                          BGNTST
5171  020432                          T29::
5172                                                          ;R1 CONTAINS BASE M8200,4,7 ADDRESS
5173  020432                                MSTCLR            ;MASTER CLEAR COMM. MICRO-PROCCESSOR FAMILY
5174  020432  004537  003156                JSR     R5,.MSTCLR          ;CLEAR M8200,4,7
5175  020436  012702  000001                MOV     #1,R2     ;START PATTERN WITH BIT0
5176  020442                                MYINT
5177  020442  013701  002716                MOV     KMCSR,R1  ;GET DEVICE ADDRESS.
5178  020446                                BGNSEG
5179  020446  104404                        TRAP    C$BSEG
5180  020450                          64$:
5181  020450  010261  000004                MOV     R2,4(R1)  ;WRITE PATTERN IN PORT4
5182  020454                                ROMCLK            ;NEXT WORD IS INSTRUCTION,  BBN
5183  020454  004537  003244                JSR     R5,.ROMCLK          ;CLOCK INSTRUCTION
5184  020460  120500                        120500            ;MOVE DATA TO THE BR REGISTER
5185  020462                                ROMCLK            ;NEXT WORD IS INSTRUCTION,  BBN
5186  020462  004537  003244                JSR     R5,.ROMCLK          ;CLOCK INSTRUCTION
5187  020466  061225                        061225            ;MOVE BR TO PORT 5
5188  020470  116104  000005                MOVB    5(R1),R4  ;PUT "FOUND" IN R4
5189  020474  120204                        CMPB    R2,R4     ;IS DATA CORRECT?
5190  020476  001414                        BEQ     65$       ;BR IF YES
5191  020500                                BERROR  3         ;DATA ERROR
5192  020520  104455                        TRAP    C$ERDF
5193  020522  000003                        .WORD   3
5194  020524  004416                        .WORD   EM3
5195  020526  006210                        .WORD   ERR3
5196  020530                          65$:  ESCAPE  SEG
5197  020530  104410                        TRAP    C$ESCAPE
5198  020532  000010                        .WORD   10000$-.
5199  020534  000241                        CLC               ;CLEAR CARRY
5200  020536  106102                        ROLB    R2        ;SHIFT BIT IN R2
5201  020540  001343                        BNE     64$       ;DONE IF R2=0
5202  020542                                ENDSEG
5203  020542                          10000$:
5204  020542  104405                        TRAP    C$ESEG
5205  020544  012702  000001                MOV     #1,R2     ;START PATTERN WITH BIT0
5206  020550                          69$:
5207  020550                                BGNSEG
5208  020550  104404                        TRAP    C$BSEG
5209  020552                          67$:
5210  020552  005102                        COM     R2
```

```
5211  020554  010261  000004                MOV     R2,4(R1)        ;WRITE PATTERN IN PORT4
5212  020560                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5213  020560  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5214  020564  120500                        120500                  ;MOVE DATA TO THE BR REGISTER
5215  020566                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5216  020566  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5217  020572  061225                        061225                  ;MOVE BR TO PORT 5
5218  020574  116104  000005                MOVB    5(R1),R4        ;PUT "FOUND" IN $GDDAT
5219  020600  010205                        MOV     R2,R5
5220  020602  120204                        CMPB    R2,R4           ;DATA CORRECT?
5221  020604  001414                        BEQ     68$             ;BR IF YES
5222  020606                                BERROR  3               ;DATA ERROR
5223  020626  104455                        TRAP    C$ERDF
5224  020630  000003                        .WORD   3
5225  020632  004416                        .WORD   EM3
5226  020634  006210                        .WORD   ERR3
5227  020636                        68$:    ESCAPE  SEG
5228  020636  104410                        TRAP    C$ESCAPE
5229  020640  000016                        .WORD   10001$-.
5230                                                                ;FAILED TO CLEAR
5231  020642  105061  000001        70$     CLRB    1(R1)           ;BRG
5232  020646  005102                        COM     R2              ;CHANGE BACK TO A ONE
5233  020650  000241                        CLC                     ;CLEAR CARRY
5234  020652  106102                        ROLB    R2              ;SHIFT BIT IN R5
5235  020654  001336                        BNE     67$             ;DONE IF R5=0
5236  020656                                ENDSEG
5237  020656                        10001$:
5238  020656  104405                        TRAP    C$ESEG
5239  020660                        ENDTST
5240  020660                        L10106:
5241  020660  104401                        TRAP    C$ETST
5242
5243  020662                                BADHEAD
5244                                        ;******************** TEST 30 ********************
5245                                        ;*SCRATCH PAD TEST
5246                                        ;*FLOAT A 1 THOUGH EACH SCRATCH PAD LOCATION
5247                                        ;*FLOAT A 0 THOUGH EACH SCRATCH PAD LOCATION
5248  020662                                BADHEAD
5249                                        ;******************** TEST 30 ********************
5250
5251  020662                        BGNTST
5252  020662                        T30::
5253  020662                                MYINT
5254  020662  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5255  020666                                MSTCLR                  ;MASTER CLEAR M8200,4,7
5256  020666  004537  003156                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
5257  020672  005002                        CLR     R2              ;START AT ADDRESS ZERO
5258  020674  012705  000001                MOV     #1,R5           ;START WITH BIT0
5259  020700                                BGNSUB
5260  020700                        T30.1:
5261  020700  104402                        TRAP    C$BSUB
5262  020702                        1$:     BGNSEG
5263  020702  104404                        TRAP    C$BSEG
5264  020704  042737  000017  020726  64$:  BIC     #17,65$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
5265  020712  050237  020726                BIS     R2,65$          ;ADD ADDRESS TO INSTRUCTION
5266  020716  010561  000004                MOV     R5,4(R1)        ;WRITE PATTERN IN PORT4
```

```
 5267  020722                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 5268  020722  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5269  020726  123100                      65$:      123100                  ;WRITE SCRATCH PAD(ADDRESS IN R2)
 5270  020730  042737  000017  020746              BIC     #17,66$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
 5271  020736  050237  020746                      BIS     R2,66$          ;ADD ADDRESS TO INSTRUCTION
 5272  020742                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 5273  020742  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5274  020746  040600                      66$:      040600                  ;MOVE SP TO BR
 5275  020750                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 5276  020750  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5277  020754  061225                                061225                  ;MOVE BR TO PORT5
 5278  020756  010537  002636                        MOV     R5,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
 5279  020762  116104  000005                        MOVB    5(R1),R4        ;PUT "FOUND" IN R4
 5280  020766  123704  002636                        CMPB    $GDDAT,R4               ;DATA CORRECT
 5281  020772  001414                                BEQ     67$             ;BR IF YES
 5282  020774                                        RERROR  4               ;DATA ERROR
 5283  021014  104455                                TRAP    C$ERDF
 5284  021016  000004                                .WORD   4
 5285  021020  004444                                .WORD   EM4
 5286  021022  006266                                .WORD   ERR4
 5287  021024                                67$:     ESCAPE  SEG
 5288  021024  104410                                TRAP    C$ESCAPE
 5289  021026  000010                                .WORD   10000$-.
 5290  021030  000241                                CLC                     ;CLEAR CARRY
 5291  021032  106105                                ROLB    R5              ;SHIFT BIT IN R5
 5292  021034  001323                                BNE     64$             ;DONE IF R5=0
 5293  021036                                        ENDSEG
 5294  021036                                10000$:
 5295  021036  104405                                TRAP    C$ESEG
 5296  021040  012705  000001                        MOV     #1,R5           ;START WITH BIT0
 5297  021044                                        BGNSEG
 5298  021044  104404                                TRAP    C$BSEG
 5299
 5300  021046  005105                      73$:      COM     R5              ;CHANGE TO FLOATING ZERO
 5301  021050  042737  000017  021072      69$:      BIC     #17,70$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
 5302  021056  050237  021072                      BIS     R2,70$          ;ADD ADDRESS TO INSTRUCTION
 5303  021062  010561  000004                        MOV     R5,4(R1)        ;WRITE PATTERN IN PORT4
 5304  021066                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 5305  021066  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5306  021072  123100                      70$:      123100                  ;WRITE SCRATCH PAD(ADDRESS IN R2)
 5307  021074  042737  000017  021112              BIC     #17,71$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
 5308  021102  050237  021112                      BIS     R2,71$          ;ADD ADDRESS TO INSTRUCTION
 5309  021106                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 5310  021106  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5311  021112  040600                      71$:      040600                  ;MOVE SP TO BR
 5312  021114                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 5313  021114  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5314  021120  061225                                061225                  ;MOVE BR TO PORT5
 5315  021122  010537  002636                        MOV     R5,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
 5316  021126  116104  000005                        MOVB    5(R1),R4        ;PUT "FOUND" IN $GDDAT
 5317  021132  123704  002636                        CMPB    $GDDAT,R4               ;DATA CORRECT?
 5318  021136  001414                                BEQ     72$             ;BR IF YES
 5319  021140                                        RERROR  4               ;DATA ERROR
 5320  021160  104455                                TRAP    C$ERDF
 5321  021162  000004                                .WORD   4
 5322  021164  004444                                .WORD   EM4
```

```
5323  021166  006266                        .WORD   ERR4
5324  021170                        72$:    ESCAPE  TST
5325  021170  104410                        TRAP    C$ESCAPE
5326  021172  000032                        .WORD   L10107-.
5327  021174  005105                        COM     R5              ;CHANGE BACK TO A ONE
5328  021176  000241                        CLC                     ;CLEAR CARRY
5329  021200  106105                        ROLB    R5              ;SHIFT BIT IN R5
5330  021202  001321                        BNE     73$             ;DONE IF R5=0
5331  021204                        ENDSEG
5332  021204                        10001$:
5333  021204  104405                        TRAP    C$ESEG
5334  021206  012705  000001                MOV     #1,R5           ;RESTART AT BIT 0
5335  021212  005202                        INC     R2              ;NEXT SP ADDRESS
5336  021214  022702  000020                CMP     #20,R2          ;LAST ADDRESS?
5337  021220  001230                        BNE     1$              ;BR IF NO
5338  021222                                ENDSUB
5339  021222                        L10110:
5340  021222  104403                        TRAP    C$ESUB
5341  021224                        ENDTST
5342  021224                        L10107:
5343  021224  104401                        TRAP    C$ETST
5344
5345  021226                                BADHEAD
5346                                        ;******************** TEST 31 ********************
5347                                        ;*SCRATCH PAD DUAL ADDRESSING TEST
5348                                        ;*WRITE AN INCREMENTING PATTERN IN ALL SP LOCATIONS
5349                                        ;*READ ALL SP LOCATIONS TO VERIFY CORRECT ADDRESSING
5350  021226                                BADHEAD
5351                                        ;******************** TEST 31 ********************
5352
5353  021226                        BGNTST
5354  021226                        T31::
5355  021226                                MSTCLR                  ;MASTER CLEAR M8200,4,7
5356  021226  004537  003156                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
5357  021232  012705  000001                MOV     #1,R5           ;START WITH A 1
5358  021236  005003                        CLR     R3              ;ADDRESS 0
5359  021240                                MYINT
5360  021240  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5361  021244                                BGNSEG
5362  021244  104404                        TRAP    C$BSEG
5363  021246  010302                1$:     MOV     R3,R2           ;MOVE ADDRESS TO R2
5364  021250  042737  000017  021272        BIC     #17,2$          ;CLEAR ADDRESS FIELD
5365  021256  050237  021272                BIS     R2,2$           ;ADD ADDRESS TO INSTRUCTION
5366  021262  010561  000004                MOV     R5,4(R1)        ;WRITE PATTERN IN PORT4
5367  021266                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5368  021266  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5369  021272  123100                2$:     123100                  ;WRITE SP(ADDRESS IN R2)
5370  021274  042737  000017  021312        BIC     #17,3$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
```

```
 5371  021302  050237  021312                BIS     R2,3$           ;ADD ADDRESS TO INSTRUCTION
 5372  021306                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 5373  021306  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
```

```
5374  021312  060600                  3$:     60600                   ;MOVE SP TO BR
5375  021314                          ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
5376  021314  004537  003244          JSR     R5,..ROMCLK                     ;CLOCK INSTRUCTION
5377  021320  061225                  61225                           ;MOVE BR TO PORT5
5378  021322  010537  002636          MOV     R5,$GDDAT                       ;PUT "EXPECTED" IN $GDDAT
5379  021326  116104  000005          MOVB    5(R1),R4                ;PUT "FOUND" IN R4
5380  021332  123704  002636          CMPB    $GDDAT,R4                       ;DATA CORRECT
5381  021336  001414                  BEQ     4$                      ;BR IF YES
5382  021340                          RERROR  4                       ;DATA ERROR
5383  021360  104455                  TRAP    C$ERDF
5384  021362  000004                  .WORD   4
5385  021364  004444                  .WORD   EM4
5386  021366  006266                  .WORD   ERR4
5387  021370                  4$:     ESCAPE  SEG
5388  021370  104410                  TRAP    C$ESCAPE
5389  021372  000014                  .WORD   10000$-.
5390  021374  005205                  INC     R5                      ;INCREMENT PATTERN
5391  021376  005203                  INC     R3                      ;NEXT ADDRESS
5392  021400  022703  000020          CMP     #20,R3                  ;LAST ADDRESS DONE?
5393  021404  001320                  BNE     1$                      ;BR IF NO
5394  021406                          ENDSEG
5395  021406                  10000$:
5396  021406  104405                  TRAP    C$ESEG
5397  021410  012705  000001          MOV     #1,R5                   ;RESTART PATTERN AT 1
5398  021414  005003                  CLR     R3                      ;RESTART AT ADDRESS ZERO
5399  021416                          BGNSEG
5400  021416  104404                  TRAP    C$BSEG
5401  021420  010302                  MOV     R3,R2           5$:     ;PUT ADDRESS IN R2
5402  021422  042737  000017  021440  BIC     #17,6$          69$:    ;CLEAR ADDRESS FIELD OF INSTRUCTION
5403  021430  050237  021440          BIS     R2,6$                   ;ADD ADDRESS TO INSTRUCTION
5404  021434                          ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
5405  021434  004537  003244          JSR     R5,..ROMCLK                     ;CLOCK INSTRUCTION
5406  021440  060600                  60600           6$:     ;MOV SP TO BR
5407  021442                          ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
5408  021442  004537  003244          JSR     R5,..ROMCLK                     ;CLOCK INSTRUCTION
5409  021446  061225                  61225                           ;MOV BR TO PORT5
5410  021450  010537  002636          MOV     R5,$GDDAT                       ;PUT "EXPECTED" IN $GDDAT
5411  021454  116104  000005          MOVB    5(R1),R4                ;PUT "FOUND" IN $GDDAT
5412  021460  123704  002636          CMPB    $GDDAT,R4                       ;DATA CORRECT?
5413  021464  001414                  BEQ     7$                      ;BR IF YES
5414  021466                          RERROR  5                       ;SP ADDRESSING ERROR
5415  021506  104455                  TRAP    C$ERDF
5416  021510  000005                  .WORD   5
5417  021512  004472                  .WORD   EM5
5418  021514  006350                  .WORD   ERR5
5419  021516                  7$:     ESCAPE  SEG
5420  021516  104410                  TRAP    C$ESCAPE
5421  021520  000014                  .WORD   10001$-.
5422  021522  005205                  INC     R5                      ;INCREMENT PATTERN
5423  021524  005203                  INC     R3                      ;NEXT ADDRESS
5424  021526  022703  000020          CMP     #20,R3                  ;LAST ADDRESS DONE?
5425  021532  001332                  BNE     5$                      ;BR IF NO
5426  021534                          ENDSEG
5427  021534                  10001$:
5428  021534  104405                  TRAP    C$ESEG
5429  021536                  ENDTST
```

```
5430  021536                          L10111:
5431  021536  104401                          TRAP    C$ETST
5432
5433  021540                                  BADHEAD
5434                                          ;******************** TEST 32 ********************
5435                                          ;*INTERRUPT TEST
5436                                          ;*TEST THAT DEVICE CAN INTERRUPT TO VECTOR A
5437  021540                                  BADHEAD
5438                                          ;******************** TEST 32 ********************
5439
5440  021540                          BGNTST
5441  021540                          T32::
5442  021540                                  MYINT
5443  021540  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5444  021544                                  BRESET                  ;BUS RESET
5445  021544  104433                          TRAP    C$RESET
5446  021546  005011                          CLR     (R1)            ;CLEAR RUN
5447  021550  004537  003552                  JSR     R5,SETVEC       ;SET UP VECTORS
5448  021554  021674                          3$                      ;XX0
5449  021556  021646                          2$                      ;XX4
5450  021560  000340  000340                  .WORD   340,340         ;LEVEL 7
5451  021564                          1$:     SETPRI  #PRI07          ;PS = LEVEL 7
5452  021564  012700  000340                  MOV     #PRI07,R0
5453  021570  104441                          TRAP    C$SPRI
5454  021572  012761  000200  000004          MOV     #200,4(R1)      ;WRITE PORT4
5455  021600                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5456  021600  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5457  021604  121111                          121111                  ;SET BR RQ IN IBUS* REG 11
5458  021606                                  SETPRI  #PRI00          ;ALLOW INTERRUPT
5459  021606  012700  000000                  MOV     #PRI00,R0
5460  021612  104441                          TRAP    C$SPRI
5461  021614  000240                          NOP
5462  021616                                  ERROR   31              ;NO INTERRUPT
5463  021634  104455                          TRAP    C$ERDF
5464  021636  000037                          .WORD   31
5465  021640  005312                          .WORD   EM31
5466  021642  010144                          .WORD   ERR31
5467  021644  000415                          BR      4$
5468  021646                          2$:     ERROR   32              ;WRONG VECTOR
5469  021664  104455                          TRAP    C$ERDF
5470  021666  000040                          .WORD   32
5471  021670  005341                          .WORD   EM32
5472  021672  010172                          .WORD   ERR32
5473  021674  062706  000004          3$:     ADD     #4,SP   ;RESET STACK
5474  021700                          4$:
5475  021700                          ENDTST
5476  021700                          L10112:
5477  021700  104401                          TRAP    C$ETST
5478
5479  021702                                  BADHEAD
5480                                          ;******************** TEST 33 ********************
5481                                          ;*INTERRUPT TEST
5482                                          ;*TEST THAT DEVICE CAN INTERRUPT TO VECTOR B
5483  021702                                  BADHEAD
5484                                          ;******************** TEST 33 ********************
5485
```

```
 5486  021702                                BGNTST
 5487  021702                                T33::
 5488  021702                                      MYINT
 5489  021702  013701  002716                      MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 5490  021706                                      MSTCLR                  ;MASTER CLEAR M8200,4,7
 5491  021706  004537  003156                      JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 5492  021712  004537  003552                      JSR     R5,SETVEC       ;SET UP VECTORS
 5493  021716  022010                              2$                      ;XX0
 5494  021720  022036                              3$                      ;XX4
 5495  021722  000340  000340                      .WORD   340,340         ;LEVEL 7
 5496  021726                                1$:   SETPRI  #PRI07          ;PS = LEVEL 7
 5497  021726  012700  000340                      MOV     #PRI07,R0
 5498  021732  104441                              TRAP    C$SPRI
 5499  021734  012761  000300  000004              MOV     #300,4(R1)      ;WRITE PORT4
 5500  021742                                      ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 5501  021742  004537  003244                      JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 5502  021746  121111                              121111                  ;SET BR RQ IN IBUS* REG 11
 5503  021750                                      SETPRI  #PRI00          ;ALLOW INTERRUPT
 5504  021750  012700  000000                      MOV     #PRI00,R0
 5505  021754  104441                              TRAP    C$SPRI
 5506  021756  000240                              NOP
 5507  021760                                      ERROR   31              ;NO INTERRUPT
 5508  021776  104455                              TRAP    C$ERDF
 5509  022000  000037                              .WORD   31
 5510  022002  005312                              .WORD   EM31
 5511  022004  010144                              .WORD   ERR31
 5512  022006  000415                              BR      4$
 5513  022010                                2$:   ERROR   32              ;WRONG VECTOR
 5514  022026  104455                              TRAP    C$ERDF
 5515  022030  000040                              .WORD   32
 5516  022032  005341                              .WORD   EM32
 5517  022034  010172                              .WORD   ERR32
 5518  022036  062706  000004            3$:   ADD     #4,SP   ;RESET STACK
 5519  022042                            4$:
 5520  022042                            ENDTST
 5521  022042                            L10113:
 5522  022042  104401                              TRAP    C$ETST
 5523
 5524  022044                                      BADHEAD
 5525                                              ;******************** TEST 34 ********************
 5526                                              ;*PRIORITY INTERRUPT TEST
 5527                                              ;*SET PS TO ALL BR LEVELS EQUAL OR GREATER THAN
 5528                                              ;*THE M8200,4,7 LEVEL, VERIFY THAT M8200,4,7 DOES NOT INTERRUPT
 5529  022044                                      BADHEAD
 5530                                              ;******************** TEST 34 ********************
 5531
 5532  022044                            BGNTST
 5533  022044                            T34::
 5534  022044                                      MYINT
 5535  022044  013701  002716                      MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 5536  022050                                      MSTCLR                  ;MASTER CLEAR M8200,4,7
 5537  022050  004537  003156                      JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 5538  022054  012704  000340                      MOV     #340,R4         ;PUT LEVEL 7 IN R2
 5539  022060                                      SETPRI  R4              ;SET PRIORITY TO 7
 5540  022060  010400                              MOV     R4,R0
 5541  022062  104441                              TRAP    C$SPRI
```

```
5542  022064  013705  002700                    MOV     STAT1,R5        ;GET BR LEVEL OF M8200,4,7
5543  022070  006205                            ASR     R5              ;SHIFT R5 4 TIMES
5544  022072  006205                            ASR     R5              ;TO GET PROPER LEVEL
5545  022074  006205                            ASR     R5
5546  022076  006205                            ASR     R5
5547  022100  042705  177437                    BIC     #177437,R5      ;CLEAR UNWANTED BITS
5548  022104  010537  002636                    MOV     R5,$GDDAT
5549  022110  004537  003552                    JSR     R5,SETVEC       ;SET UP VECTORS
5550  022114  022160                            2$                      ;A VECTOR
5551  022116  022160                            2$                      ;B VECTOR
5552  022120  000340  000340                    .WORD   340,340         ;PRIORITY 7
5553  022124  012761  000200  000004  4$:      MOV     #200,4(R1)      ;LOAD PORT4
5554  022132                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5555  022132  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5556  022136  121111                            121111                  ;SET BR REQUEST
5557  022140                            5$:     SETPRI  R4              ;PUT LEVEL IN R2 IN PS
5558  022140  010400                            MOV     R4,R0
5559  022142  104441                            TRAP    C$SPRI
5560  022144  000240                            NOP
5561  022146  020504                            CMP     R5,R4           ;IS PRESENT PS LEVEL = TO M8200,4,7 LEVEL
5562  022150  001420                            BEQ     1$              ;BR IF YES
5563  022152  162704  000040                    SUB     #40,R4          ;NO GET NEXT LOWER LEVEL IN R2
5564  022156  000770                            BR      5$              ;AND CONTINUE WITH TEST
5565  022160                            2$:     BRESET
5566  022160  104433                            TRAP    C$RESET
5567  022162                                    ERROR   33              ;ERROR UNEXPECTED INTERRUPT
5568  022200  104455                            TRAP    C$ERDF
5569  022202  000041                            .WORD   33
5570  022204  005400                            .WORD   EM33
5571  022206  010220                            .WORD   ERR33
5572  022210  000002                            RTI
5573  022212                            1$:     MSTCLR
5574  022212  004537  003156                    JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
5575  022216                                    ENDTST
5576  022216                            L10114:
5577  022216  104401                            TRAP    C$ETST
5578
5579  022220                                    BADHEAD
5580                                            ;******************** TEST 35 ********************
5581                                            ;*PRIORITY INTERRUPT TESTS
5582                                            ;*SET PS TO ALL BR LEVELS LESS THAN THE M8200,4,7 LEVEL
5583                                            ;*VERIFY THAT M8200,4,7 WILL INTERRUPT
5584  022220                                    BADHEAD
5585                                            ;******************** TEST 35 ********************
5586
5587  022220                            BGNTST
5588  022220                            T35::
5589  022220                                    MYINT
5590  022220  013701  002716                    MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5591  022224                                    MSTCLR                  ;MASTER CLEAR M8200,4,7
5592  022224  004537  003156                    JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
5593  022230  012704  000340                    MOV     #340,R4         ;PUT LEVEL 7 IN R2
5594  022234                                    SETPRI  R4              ;SET PRIORITY TO 7
5595  022234  010400                            MOV     R4,R0
5596  022236  104441                            TRAP    C$SPRI
5597  022240  013705  002700                    MOV     STAT1,R5        ;GET BR LEVEL OF M8200,4,7
```

```
 5598  022244  006205                        ASR     R5              ;SHIFT R5 4 TIMES
 5599  022246  006205                        ASR     R5              ;TO GET PROPER LEVEL
 5600  022250  006205                        ASR     R5
 5601  022252  006205                        ASR     R5
 5602  022254  042705  177437                BIC     #177437,R5      ;CLEAR UNWANTED BITS
 5603  022260  010502                        MOV     R5,R2           ;PUT M8200,4,7 LEVEL IN R2
 5604  022262  162702  000040                SUB     #40,R2          ;GET NEXT LOWER LEVEL IN R2
 5605  022266  004537  003552                JSR     R5,SETVEC       ;SET UP VECTORS
 5606  022272  022354                        2$                      ;A VECTOR
 5607  022274  022362                        3$                      ;B VECTOR
 5608  022276  000340  000340                .WORD   340,340         ;PRIORITY 7
 5609  022302  012761  000200  000004  4$:   MOV     #200,4(R1)      ;LOAD PORT4
 5610  022310                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 5611  022310  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5612  022314  121111                        121111                  ;SET BR REQUEST
 5613  022316                        5$:     SETPRI  R2              ;PUT LEVEL IN R2 IN PS
 5614  022316  010200                        MOV     R2,R0
 5615  022320  104441                        TRAP    C$SPRI
 5616  022322  000240                        NOP
 5617  022324                                ERROR   31              ;ERROR, NO INTERRUPT
 5618  022342  104455                        TRAP    C$ERDF
 5619  022344  000037                        .WORD   31
 5620  022346  005312                        .WORD   EM31
 5621  022350  010144                        .WORD   ERR31
 5622  022352  000421                6$:     BR      1$
 5623  022354  012716  022352        2$:     MOV     #6$,(SP)        ;SET UP FOR RTI
 5624  022360  000002                        RTI
 5625  022362                        3$:     ERROR   32              ;ERROR, WRONG VECTOR
 5626  022400  104455                        TRAP    C$ERDF
 5627  022402  000040                        .WORD   32
 5628  022404  005341                        .WORD   EM32
 5629  022406  010172                        .WORD   ERR32
 5630  022410  012716  022416                MOV     #1$,(SP)        ;SET UP FOR RTI
 5631  022414  000002                        RTI
 5632  022416                        1$:     MSTCLR
 5633  022416  004537  003156                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 5634  022422                                ENDTST
 5635  022422                        L10115:
 5636  022422  104401                        TRAP    C$ETST
 5637
 5638  022424                                BADHEAD
 5639                                        ;******************** TEST 36 ********************
 5640                                        ;*NPR TEST
 5641                                        ;*TEST OF DATO, 1 WORD FROM UPROC TO 11 MEMORY
 5642  022424                                BADHEAD
 5643                                        ;******************** TEST 36 ********************
 5644
 5645  022424                        BGNTST
 5646  022424                        T36::
 5647  022424                                BRESET                  ;BUS RESET
 5648  022424  104433                        TRAP    C$RESET
 5649
 5650  022426                                MYINT
 5651  022426  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 5652  022432  005011                        CLR     (R1)            ;CLEAR RUN
 5653  022434  005061  000004                CLR     4(R1)           ;CLR PORT4
```

```
5654  022440  004537  003574          JSR     R5,NPRSET       ;SET UP IBUS REG 0-7
5655  022444  000000                  0                       ;IN DATA
5656  022446  177777                  -1                      ;OUT DATA
5657  022450  022564                  3$                      ;IN BA
5658  022452  022562                  2$                      ;OUT BA
5659  022454  005037  022562          CLR     2$              ;CLEAR 2$
5660  022460  005061  000004          CLR     4(R1)           ;CLEAR PORT 4
5661  022464                          ROMCLK                  ;NOW MOVE TO IBUS*<11>
5662  022464  004537  003244          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5663  022470  121111                  121111
5664  022472  012761  000021  000004  MOV     #21,4(R1)       ;WRITE PORT4
5665  022500                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5666  022500  004537  003244          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5667  022504  121110                  121110                  ;SET NPR BITS IN IBUS* REG 10
5668  022506  000240                  NOP
5669  022510  012737  177777  002636  MOV     #-1,$GDDAT              ;PUT "EXPECTED" IN $GDDAT
5670  022516  013704  022562          MOV     2$,R4           ;PUT "FOUND" IN R4
5671  022522  023704  002636          CMP     $GDDAT,R4               ;DATA CORRECT?
5672  022526  001413                  BEQ     4$              ;BR IF YES
5673  022530                          ERROR   11,YES          ;ERROR NPR FAILED
5674  022542  104455                  TRAP    C$ERDF
5675  022544  000013                  .WORD   11
5676  022546  004654                  .WORD   EM11
5677  022550  006660                  .WORD   ERR11
5678  022552                          ESCAPE  TST
5679  022552  104410                  TRAP    C$ESCAPE
5680  022554  000012                  .WORD   L10116-.
5681  022556                  4$:     EXIT    TST
5682  022556  104432                  TRAP    C$EXIT
5683  022560  000006                  .WORD   L10116-.
5684  022562  000000                  2$:     0                       ;OUT BA
5685  022564  000000                  3$:     0                       ;IN BA
5686  022566                          ENDTST
5687  022566                          L10116:
5688  022566  104401                  TRAP    C$ETST
5689
5690  022570                          BADHEAD
5691                                  ;******************** TEST 37 ********************
5692                                  ;*NPR TEST
5693                                  ;*TEST OF DATI, 1 WORD FROM 11 MEMORY TO UPROC
5694  022570                          BADHEAD
5695                                  ;******************** TEST 37 ********************
5696
5697  022570                  BGNTST
5698  022570                  T37::
5699  022570                          MYINT
5700  022570  013701  002716          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5701  022574                          MSTCLR                  ;MASTER CLEAR M8200,4,7
5702  022574  004537  003156          JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
5703  022600  005061  000004          CLR     4(R1)           ;CLR PORT4
5704  022604  004537  003574          JSR     R5,NPRSET       ;SET UP IBUS REG 0-7
5705  022610  000000                  0                       ;IN DATA
5706  022612  177777                  -1                      ;OUT DATA
5707  022614  022734                  3$                      ;IN BA
5708  022616  022732                  2$                      ;OUT BA
5709  022620  012737  177777  022734  MOV     #-1,3$          ;PUT DATA IN 3$
```

```
5710  022626  012761  000001  000004          MOV     #1,4(R1)        ;WRITE PORT4
5711  022634                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5712  022634  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5713  022640  121110                          121110                  ;SET NPR BITS IN IBUS* REG 11
5714  022642  000240                          NOP
5715  022644  012737  177777  002636          MOV     #-1,$GDDAT              ;PUT "EXPECTED" IN $GDDAT
5716  022652                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5717  022652  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5718  022656  021004                          021004                  ;MOVE IN DATA LOW BYTE TO PORT4
5719  022660                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5720  022660  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5721  022664  021025                          021025                  ;MOVE IN DATA HIGH BYTE TO PORT5
5722  022666  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
5723  022672  023704  002636                  CMP     $GDDAT,R4               ;DATA CORRECT?
5724  022676  001413                          BEQ     4$              ;BR IF YES
5725  022700                                  ERROR   11,YES          ;ERROR NPR FAILED
5726  022712  104455                          TRAP    C$ERDF
5727  022714  000013                          .WORD   11
5728  022716  004654                          .WORD   EM11
5729  022720  006660                          .WORD   ERR11
5730  022722                                  ESCAPE  TST
5731  022722  104410                          TRAP    C$ESCAPE
5732  022724  000012                          .WORD   L10117-.
5733  022726                          4$:     EXIT    TST
5734  022726  104432                          TRAP    C$EXIT
5735  022730  000006                          .WORD   L10117-.
5736  022732  000000                  2$:     0                       ;OUT BA
5737  022734  000000                  3$:     0                       ;IN BA
5738  022736                          ENDTST
5739  022736                          L10117:
5740  022736  104401                          TRAP    C$ETST
5741
5742  022740                                  BADHEAD
5743                                          ;******************** TEST 38 ********************
5744                                          ;*NPR TEST
5745                                          ;*TEST OF DATOB, 1 BYTE FROM UPROC TO 11 MEMORY
5746  022740                                  BADHEAD
5747                                          ;******************** TEST 38 ********************
5748
5749  022740                          BGNTST
5750  022740                          T38::
5751  022740                                  MYINT
5752  022740  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5753  022744                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
5754  022744  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
5755  022750  005061  000004                  CLR     4(R1)           ;CLR PORT4
5756  022754  004537  003574                  JSR     R5,NPRSET       ;SET UP IBUS REG 0-7
5757  022760  000000                          0                       ;IN DATA
5758  022762  177777                          -1                      ;OUT DATA
5759  022764  023100                          3$                      ;IN BA
5760  022766  023077                          2$+1                    ;OUT BA
5761  022770  005037  023076                  CLR     2$              ;CLEAR 2$
5762  022774  005061  000004                  CLR     4(R1)           ;CLEAR PORT 4
5763  023000                                  ROMCLK                  ;NOW MOVE IT TO IBUS*<11>
5764  023000  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5765  023004  121111                          121111
```

```
 5766  023006  012761  00C221  000004          MOV     #221,4(R1)        ;WRITE PORT4
 5767  023014                                  ROMCLK                    ;NEXT WORD IS INSTRUCTION,  BBN
 5768  023014  004537  003244                  JSR     R5,..ROMCLK               ;CLOCK INSTRUCTION
 5769  023020  121110                          121110                    ;SET NPR BITS IN IBUS* REG 11
 5770  023022  000240                          NOP
 5771  023024  012737  177400  002636          MOV     #177400,$GDDAT    ;PUT "EXPECTED" IN $GDDAT
 5772  023032  013704  023076                  MOV     2$,R4             ;PUT "FOUND" IN R4
 5773  023036  023704  002636                  CMP     $GDDAT,R4                 ;DATA CORRECT?
 5774  023042  001413                          BEQ     4$                ;BR IF YES
 5775  023044                                  ERROR   11,YES            ;ERROR NPR FAILED
 5776  023056  104455                          TRAP    C$ERDF
 5777  023060  000013                          .WORD   11
 5778  023062  004654                          .WORD   EM11
 5779  023064  006660                          .WORD   ERR11
 5780  023066                                  ESCAPE  TST
 5781  023066  104410                          TRAP    C$ESCAPE
 5782  023070  000012                          .WORD   L10120-.
 5783  023072                          4$:     EXIT    TST
 5784  023072  104432                          TRAP    C$EXIT
 5785  023074  000006                          .WORD   L10120-.
 5786  023076  000000                  2$:     0                         ;OUT BA
 5787  023100  000000                  3$:     0                         ;IN BA
 5788  023102                          ENDTST
 5789  023102                          L10120:
 5790  023102  104401                          TRAP    C$ETST
 5791
 5792  023104                                  BADHEAD
 5793                                          ;******************** TEST 39 ********************
 5794                                          ;*TEST OF EA BITS 16 AND 17
 5795                                          ;*DO A DATO TO AN ADDRESS USING OUT BA BITS 16 AND 17
 5796                                          ;*VERIFY CORRECT RESULTS
 5797  023104                                  BADHEAD
 5798                                          ;******************** TEST 39 ********************
 5799
 5800  023104                          BGNTST
 5801  023104                          T39::
 5802  023104                                  MSTCLR                    ;MASTER CLEAR M8200,4,7
 5803  023104  004537  003156                  JSR     R5,..MSTCLR               ;CLEAR M8200,4,7
 5804  023110                                  MYINT
 5805  023110  013701  002716                  MOV     KMCSR,R1          ;GET DEVICE ADDRESS.
 5806  023114  013737  002726  023142          MOV     KMP06,1$          ;USE SEL4 FOR ADDRESS
 5807  023122  013737  002726  023140          MOV     KMP06,2$          ;USE SEL4 FOR ADDRESS
 5808  023130  004537  003574                  JSR     R5,NPRSET         ;LOAD BA AND DATA
 5809  023134  000000                          0                         ;IN DATA
 5810  023136  125252                          125252                    ;OUT DATA
 5811  023140  000000                  2$:     0                         ;IN BA
 5812  023142  000000                  1$:     0                         ;OUT BA
 5813  023144  012761  000014  000004          MOV     #14,4(R1)         ;LOAD SEL 4 WITH OUT BA16 AND 17
 5814  023152                                  ROMCLK                    ;NEXT WORD IS INSTRUCTION,  BBN
 5815  023152  004537  003244                  JSR     R5,..ROMCLK               ;CLOCK INSTRUCTION
 5816  023156  121111                          121111                    ;SET OUTBA 16 AND 17
 5817  023160  012761  000021  000004          MOV     #21,4(R1)         ;LOAD SEL4
 5818  023166  012711  003000                  MOV     #BIT9!BIT10,(R1)
 5819  023172  012761  121110  000006          MOV     #121110,6(R1)     ;PUT INSTRUCTION IN SEL6
 5820  023200  052711  000400                  BIS     #BIT8,(R1)         ;CLOCK IT!
 5821  023204  000240                          NOP                       ;WAIT FOR NPR
```

```
5822  023206  012737  121110  002636          MOV     #121110,$GDDAT  ;PUT 'EXPECTED' IN $GDDAT
5823  023214  000240                          NOP
5824  023216  000240                          NOP
5825                                                                  ;OK,LISTEN UP!EXPLAINATION TIME.
5826                                                                  ;
5827                                                                  ;ON THE NPR OUT,THE DATA ENDED UP
5828                                                                  ;IN THE IBUS(NOT IBUS*) SENCE SEL A
5829                                                                  ;WAS ONLY SELECTED IN THE NPR CYCLE.
5830                                                                  ;THAT IS,WE DIDN'T REALLY DO AN NPR TO
5831                                                                  ;PORT 6,THE NPR OUT REALLY ENDED UP IN
5832                                                                  ;OUT DATA LOW,AND OUT DATA HIGH
5833                                                                  ;(IBUS <2> AND IBUS <3>).
5834
5835                                                                  ;WHAT WE'RE DOING NEXT IS READING IBUS 2&3
5836                                                                  ;TO SEE IF THE DATA GOT XFERRED CORRECTLY.
5837  023220                                  ROMCLK
5838  023220  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5839  023224  021044                          021044                  ;READ IBUS <2> PUT IN PORT 4
5840  023226                                  ROMCLK
5841  023226  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5842  023232  021065                          021065                  ;READ IBUS <3> PUT IN PORT 5
5843  023234  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
5844  023240  023704  002636                  CMP     $GDDAT,R4               ;CORRECT RESULTS?
5845  023244  001411                          BEQ     3$              ;BR IF YES
5846  023246                                  ERROR   11,YES          ;ERROR BA 16 AND 17 FAILED
5847  023260  104455                          TRAP    C$ERDF
5848  023262  000013                          .WORD   11
5849  023264  004654                          .WORD   EM11
5850  023266  006660                          .WORD   ERR11
5851  023270                          3$:
5852  023270                          ENDTST
5853  023270                          L10121:
5854  023270  104401                          TRAP    C$ETST
5855
5856  023272                                  BADHEAD
5857                                          ;******************** TEST 40 ********************
5858                                          ;*TEST OF EA BITS 16 AND 17
5859                                          ;*DO A DATI USING IN BA BITS 16 AND 17
5860                                          ;*VERIFY CORRECT RESULTS
5861                                          ;*IN ORDER TO DO THIS TEST, WE WILL READ THE DATA FROM THE
5862                                          ;*CONSOL TTY CSR IF ONE EXSITS
5863                                          ;*IF NO CONSOL TTY CSR AT ADDRESS 177560, THIS TEST
5864                                          ;*WILL BE SKIPPED
5865  023272                                  BADHEAD
5866                                          ;******************** TEST 40 ********************
5867
5868  023272                          BGNTST
5869  023272                          T40::
5870  023272                                  MYINT
5871  023272  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5872  023276                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
5873  023276  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
5874  023302  012737  023464  000004  STOP:   MOV     #TOUTT,4        ;SET UP FOR TRAP IN CASE IF NO
5875  023310  012737  000340  000006          MOV     #340,6  ;TTY AT ADDRESS 177560
5876  023316  005737  177560                  TST     177560          ;ADDRESS THE TTY-TRAPS HERE IF NONE.
5877  023322  012737  177560  023350          MOV     #177560,1$      ;USE SEL4 FOR ADDRESS
```

```
5878  023330  012737  177560  023346          MOV     #177560,2$      ;USE SEL4 FOR ADDRESS
5879  023336  004537  003574                  JSR     R5,NPRSET       ;LOAD BA AND DATA
5880  023342  000000                          0                       ;IN DATA
5881  023344  125252                          125252                  ;OUT DATA
5882  023346  000000                  2$:     0                       ;IN BA
5883  023350  000000                  1$:     0                       ;OUT BA
5884  023352  012761  000015  000004          MOV     #15,4(R1)
5885  023360  012711  003000                  MOV     #BIT9!BIT10,(R1);SET CROMI AND CROMO!!
5886  023364  012761  121110  000006          MOV     #121110,6(R1)   ;PUT INSTR INTO SEL6  NW*
5887  023372  052711  000400                  BIS     #BIT8,(R1)      ;CLOCK IT!
5888  023376  000240                          NOP                     ;WAIT FOR NPR
5889  023400                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5890  023400  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5891  023404  021004                          021004                  ;MOVE OUT DATA LB TO SEL4
5892  023406                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5893  023406  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5894  023412  021025                          021025                  ;MOVE OUT DATA HB TO SEL5
5895  023414  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
5896  023420  013737  177560  002636          MOV     177560,$GDDAT
5897  023426  042737  000200  002636          BIC     #200,$GDDAT
5898  023434  023704  002636                  CMP     $GDDAT,R4               ;CORRECT RESULTS?
5899  023440  001413                          BEQ     TOUTP           ;BR IF YES
5900  023442                                  ERROR   11,YES          ;ERROR BA 16 AND 17 FAILED
5901  023454  104455                          TRAP    C$ERDF
5902  023456  000013                          .WORD   11
5903  023460  004654                          .WORD   EM11
5904  023462  006660                          .WORD   ERR11
5905  023464                          3$:
5906  023464  062706  000004          TOUTT:  ADD     #4,SP           ;UPDATE STACK POITNTER
5907  023470  013737  002652  000006  TOUTP:  MOV     SAVE6,6         ;RESTORE TRAP VECTOR
5908  023476  013737  002650  000004          MOV     SAVE4,4
5909  023504                          ENDTST
5910  023504                          L10122:
5911  023504  104401                          TRAP    C$ETST
5912
5913  023506                                  BADHEAD
5914                                          ;******************** TEST 41 ********************
```

```
 5915                                                ;*NPR NON-EXISTENT MEMORY TEST
 5916                                                ;*DO A DATO TO A NON-EXISTENT ADDRESS
 5917                                                ;*VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
 5918  023506                                        BADHEAD
 5919                                                ;******************** TEST 41 ********************
 5920
 5921  023506                                BGNTST
 5922  023506                                T41::
 5923  023506                                        MYINT
 5924  023506  013701  002716                       MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 5925  023512                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
 5926  023512  004537  003156                       JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 5927  023516  004537  003574                       JSR     R5,NPRSET       ;LOAD IBUS REGISTERS 0-7
 5928  023522  000000                               0                       ;IN DATA
 5929  023524  000000                               0                       ;OUT DATA
 5930  023526  177320                               177320                  ;IN BA
 5931  023530  177320                               177320                  ;IN BA
 5932  023532  012761  000014  000004               MOV     #14,4(R1)       ;SET OUT BA BITS 16+17 IN PORT4
```

```
 5933  023540                                    ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
 5934  023540  004537  003244                    JSR     R5,.ROMCLK                      ;CLOCK INSTRUCTION
 5935  023544  121111                            121111                          ;SET OUTBA 16 AND 17
 5936  023546  012761  000021  000004            MOV     #21,4(R1)               ;SET NPR REQUEST BITS IN PORT4
 5937  023554                                    ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
 5938  023554  004537  003244                    JSR     R5,.ROMCLK                      ;CLOCK INSTRUCTION
 5939  023560  121110                            121110                          ;MOV IBUS* 4 TO IBUS* 10
 5940  023562  000240                            NOP
 5941  023564                                    ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
 5942  023564  004537  003244                    JSR     R5,.ROMCLK                      ;CLOCK INSTRUCTION
 5943  023570  121225                            121225                          ;MOV IBUS*11 TO IBUS*5
 5944  023572  012737  000001  002636            MOV     #1,$GDDAT                       ;PUT "EXPECTED" IN $GDDAT
 5945  023600  116104  000005                    MOVB    5(R1),R4                ;PUT "FOUND" IN R4
 5946  023604  042704  177776                    BIC     #177776,R4              ;CLEAR UNWANTED BITS
 5947  023610  023704  002636                    CMP     $GDDAT,R4                       ;DATA CORRECT?
 5948  023614  001411                            BEQ     1$                      ;BR IF YES
 5949  023616                                    ERROR   13,YES                  ;ERROR NON-EXISTENT MEM BIT FAILED TO SET
 5950  023630  104455                            TRAP    C$ERDF
 5951  023632  000015                            .WORD   13
 5952  023634  004707                            .WORD   EM13
 5953  023636  007014                            .WORD   ERR13
 5954  023640                            1$:
 5955  023640  152761  000100  000001            BISB    #100,1(R1)              ;SET MASTER CLEAR
 5956  023646  142761  000100  000001            BICB    #100,1(R1)              ;CLEAR MASTER
 5957  023654                                    ROMCLK                          ;MOV IBUS*11 TO
 5958  023654  004537  003244                    JSR     R5,.ROMCLK                      ;CLOCK INSTRUCTION
 5959  023660  121225                            121225                          ;PORT5
 5960  023662  005037  002636                    CLR     $GDDAT                  ;EXPECT CLEAR
 5961  023666  116104  000005                    MOVB    5(R1),R4                ;GET NPR REG
 5962  023672  042704  177776                    BIC     #177776,R4              ;CLEAR JUNK
 5963  023676  001411                            BEQ     2$                      ;EXIT IF CLEAR
 5964  023700                                    ERROR   13,YES                  ;NON-EXISTANT MEM
 5965  023712  104455                            TRAP    C$ERDF
 5966  023714  000015                            .WORD   13
 5967  023716  004707                            .WORD   EM13
 5968  023720  007014                            .WORD   ERR13
 5969                                                                            ;BIT FAILED TO CLEAR
 5970  023722                            2$:
 5971  023722                            ENDTST
 5972  023722                            L10123:
 5973  023722  104401                            TRAP    C$ETST
 5974
 5975  023724                                    BADHEAD
 5976                                            ;******************** TEST 42 ********************
 5977                                            ;*NPR NON-EXISTENT MEMORY TEST
 5978                                            ;*DO A DATI FROM A NON-EXISTENT ADDRESS
 5979                                            ;*VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
 5980  023724                                    BADHEAD
 5981                                            ;******************** TEST 42 ********************
 5982
 5983  023724                            BGNTST
 5984  023724                            T42::
 5985  023724                                    MYINT
 5986  023724  013701  002716                    MOV     KMCSR,R1                ;GET DEVICE ADDRESS.
 5987  023730                                    MSTCLR                          ;MASTER CLEAR M8200,4,7
 5988  023730  004537  003156                    JSR     R5,.MSTCLR                      ;CLEAR M8200,4,7
```

```
 5989  023734  004537  003574          JSR     R5,NPRSET       ;LOAD IBUS REGISTERS 0-7
 5990  023740  000000                  0                       ;IN DATA
 5991  023742  000000                  0                       ;OUT DATA
 5992  023744  177320                  177320                  ;IN BA
 5993  023746  177320                  177320                  ;OUT BA
 5994  023750  005061  000004          CLR     4(R1)
 5995  023754                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 5996  023754  004537  003244          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 5997  023760  121111                  121111                  ;CLEAR NON-EXISTENT BIT
 5998  023762  012761  000015  000004  MOV     #15,4(R1)       ;SET NPR REQUEST BITS IN PORT4
 5999  023770                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 6000  023770  004537  003244          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6001  023774  121110                  121110                  ;MOV IBUS* 4 TO IBUS* 10
 6002  023776  000240                  NOP
 6003  024000                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 6004  024000  004537  003244          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6005  024004  121225                  121225                  ;MOV IBUS*11 TO IBUS*5
 6006  024006  012737  000001  002636  MOV     #1,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
 6007  024014  116104  000005          MOVB    5(R1),R4        ;PUT "FOUND" IN R4
 6008  024020  042704  177776          BIC     #177776,R4      ;CLEAR UNWANTED BITS
 6009  024024  023704  002636          CMP     $GDDAT,R4               ;DATA CORRECT?
 6010  024030  001411                  BEQ     1$              ;BR IF YES
 6011  024032                          ERROR   13,YES          ;ERROR NON-EXISTENT MEM BIT FAILED TO SET
 6012  024044  104455                  TRAP    C$ERDF
 6013  024046  000015                  .WORD   13
 6014  024050  004707                  .WORD   EM13
 6015  024052  007014                  .WORD   ERR13
 6016  024054                  1$:
 6017  024054                  ENDTST
 6018  024054                  L10124:
 6019  024054  104401                  TRAP    C$ETST
 6020
 6021  024056                          BADHEAD
 6022                                  ;******************** TEST 43 ********************
 6023                                  ;*NPR TEST
 6024                                  ;*USING DATO, NPR A BINARY COUNT (0-377)
 6025                                  ;*FROM MICRO-PROCESSOR TO ALL AVAILABLE MEMORY
 6026  024056                          BADHEAD
 6027                                  ;******************** TEST 43 ********************
 6028
 6029  024056                  BGNTST
 6030  024056                  T43::
 6031  024056                          MYINT
 6032  024056  013701  002716          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6033  024062                          MSTCLR                  ;MASTER CLEAR M8200,4,7
 6034  024062  004537  003156          JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 6035  024066  005037  024270          CLR     5$              ;START FLAG AT 0
 6036  024072  005005                  CLR     R5              ;DATA
 6037  024074  012702  035404          MOV     #CORMAX,R2      ;ADDRESS
 6038  024100                  1$:
 6039  024100  010537  024130          MOV     R5,2$           ;LOAD DATA
 6040  024104  010237  024134          MOV     R2,4$           ;LOAD BA
 6041  024110  032702  000001          BIT     #BIT0,R2        ;IS BA ODD?
 6042  024114  001402                  BEQ     .+6             ;BR IF NO
 6043  024116  000337  024130          SWAB    2$              ;IF ODD PUT DATA IN HI-BYTE
 6044  024122  004537  003574          JSR     R5,NPRSET       ;LOAD NPR REGISTERS
```

```
6045  024126  000000                        0                       ;IN DATA
6046  024130  000000                2$:     0                       ;OUT DATA
6047  024132  000000                        0                       ;IN BA
6048  024134  000000                4$:     0                       ;OUT BA
6049  024136  105012                        CLRB    (R2)            ;CLEAR MEMORY LOCATION
6050  024140  012761  000221  000004       MOV     #221,4(R1)      ;LOAD PORT4
6051  024146                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
6052  024146  004537  003244               JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6053  024152  121110                        121110                  ;DO THE NPR
6054  024154  000240                        NOP
6055  024156  010537  002636               MOV     R5,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
6056  024162  111204                        MOVB    (R2),R4 ;PUT "FOUND" IN R4
6057  024164  123704  002636               CMPB    $GDDAT,R4               ;IS DATA CORRECT?
6058  024170  001411                        BEQ     3$              ;BR IF YES
6059  024172                                ERROR   11,YES          ;ERROR, DATA INCORRECT
6060  024204  104455                        TRAP    C$ERDF
6061  024206  000013                        .WORD   11
6062  024210  004654                        .WORD   EM11
6063  024212  006660                        .WORD   ERR11
6064  024214                        3$:     ESCAPE  TST
6065  024214  104410                        TRAP    C$ESCAPE
6066  024216  000054                        .WORD   L10125-.
6067  024220  005205                        INC     R5              ;NEXT CHARACTER
6068  024222  042705  177400               BIC     #177400,R5      ;USE ONLY LOW BYTE
6069  024226  005737  024270               TST     5$              ;HAS MAX MEMORY BEEN REACHED YET?
6070  024232  001402                        BEQ     6$              ;BR IF NO
6071  024234  005705                        TST     R5              ;DONE PATTERN?
6072  024236  001412                        BEQ     7$              ;BR IF YES
6073  024240  005202                6$:     INC     R2              ;INC BA
6074  024242  023702  002604               CMP     MEMLIM,R2       ;REACHED MEMORY LIMIT YET?
6075  024246  001314                        BNE     1$              ;BR IF NOT
6076  024250  012702  035404               MOV     #CORMAX,R2      ;RESTART BA AT FIRST ADDRESS
6077  024254  012737  177777  024270       MOV     #-1,5$          ;SET FLAG TO END TEST AT END OF DATA PATTERN
6078  024262  000706                        BR 1$                   ;CONTINUE
6079  024264                        7$:
6080  024264                                EXIT    TST
6081  024264  104432                        TRAP    C$EXIT
6082  024266  000004                        .WORD   L10125-.
6083  024270  000000                5$:     0                       ;THIS LOCATION IS A FLAG, IT STARTS AT 0,
6084                                                                ;AND IS SET TO -1 WHEN LAST MEMORY ADDRESS
6085                                                                ;IS USED, TEST IS THEN ENDED WHEN PATTERN IS FINISHED
6086  024272                        ENDTST
6087  024272                        L10125:
6088  024272  104401                        TRAP    C$ETST
6089                                ;$MEM1
6090                                ;$MEM0
6091                                ;$MEM2  1K
6092                                ;$MEM3  1K
6093
6094  024274                                BADHEAD
6095                                ;******************** TEST 44 ********************
6096                                ;*ALU C BIT TEST
6097                                ;*TEST THAT AN ADD OF 377 AND 377 WILL SET THE C BIT
6098  024274                                BADHEAD
6099                                ;******************** TEST 44 ********************
6100
```

```
6101  024274                                BGNTST
6102  024274                                T44::
6103  024274                                        MYINT
6104  024274  013701  002716                        MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6105  024300                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
6106  024300  004537  003156                        JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
6107  024304  004737  003640                        JSR     PC,MEMLD        ;LOAD MAINMEM DATA
6108  024310  024424                                TDATA                   ;POINTER TO DATA
6109  024312  004737  004012                        JSR     PC,SPLD         ;LOAD SP DATA
6110  024316  024424                                TDATA                   ;POINTER TO DATA
6111  024320                                        BGNSEG
6112  024320  104404                                TRAP    C$BSEG
6113  024322                                1$:
6114  024322                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
6115  024322  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6116  024326  010000                                010000                  ;MAR 0
6117  024330                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
6118  024330  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6119  024334  054400                                054400!<0*20>           ;ADD 377 AND 377, TO SET C BIT
6120  024336                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
6121  024336  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6122  024342  040421                                040401!<1*20>           ;ADD 0 AND 0 AND THE C BIT
6123  024344                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
6124  024344  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6125  024350  061224                                61224                   ;PUT RESULTS IN PORT4
6126  024352  012737  000001  002636                MOV     #1,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
6127  024360  016104  000004                        MOV     4(R1),R4        ;PUT "FOUND" IN R4
6128  024364  123704  002636                        CMPB    $GDDAT,R4               ;DATA CORRECT?
6129  024370  001411                                BEQ     2$              ;BR IF YES
6130  024372                                        ERROR   34,YES          ;ERROR C BIT NOT SET
6131  024404  104455                                TRAP    C$ERDF
6132  024406  000042                                .WORD   34
6133  024410  005435                                .WORD   EM34
6134  024412  010276                                .WORD   ERR34
6135  024414                                2$:     ESCAPE  SEG
6136  024414  104410                                TRAP    C$ESCAPE
6137  024416  000002                                .WORD   10000$-.
6138  024420                                        ENDSEG
6139  024420                                10000$:
6140  024420  104405                                TRAP    C$ESEG
6141  024422                                ENDTST
6142  024422                                L10126:
6143  024422  104401                                TRAP    C$ETST
6144  024424     377     000     000        TDATA:  .BYTE   -1,0,0,0,0,0,0,0
6145  024427     000     000     000
6146  024432     000     000
6147
6148                                        .EVEN
6149
6150  024434                                        BADHEAD
6151                                                ;******************** TEST 45 ********************
6152                                                ;*ALU TEST
6153                                                ;*TEST OF ALU FUNCTION SEL B WITH C BIT CLEARED
6154                                                ;*ALU FUNCTION (B)    CODE=11
6155                                                ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6156                                                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
```

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

                                        BGNTST
                                        T45::
                                                MYINT
024434 013701 002716                            MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
024440                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
024440 004537 003156                            JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
024444 005005                                   CLR     R5              ;MEM + SP ADDRESS
024446 012702 024626                            MOV     #5$,R2          ;POINTER TO CORRECT DATA
024452 004737 003640                            JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
024456 002654                                   MEMDAT                  ;POINTER TO DATA
024460 004737 004012                            JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
024464 002664                                   SPDAT                   ;POINTER TO DATA
024466                                          BGNSEG
024466 104404                                   TRAP    C$BSEG
024470 004737 004060                    1$:     JSR     PC,CLRC         ;CLEAR C BIT!
024474 042737 000017 024512                     BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
024502 050537 024512                            BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
024506                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024506 004537 003244                            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
024512 010000                           2$:     010000                  ;LOAD MAR
024514 042737 000017 024532                     BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
024522 050537 024532                            BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
024526                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024526 004537 003244                            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
024532 040620                           3$:     040400!<11*20>          ;BR   SEL B
024534                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024534 004537 003244                            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
024540 061224                                   61224                   ;MOVE BR TO PORT4
024542 111237 002636                            MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
024546 116104 000004                            MOVB    4(R1),R4        ;PUT "FOUND" IN R4
024552 123704 002636                            CMPB    $GDDAT,R4               ;DATA CORRECT?
024556 001411                                   BEQ     4$              ;BR IF YES
024560                                          ERROR   15,YES          ;ALU ERROR
024572 104455                                   TRAP    C$ERDF
024574 000017                                   .WORD   15
024576 004754                                   .WORD   EM15
024600 007120                                   .WORD   ERR15
024602                                  4$:     ESCAPE  SEG
024602 104410                                   TRAP    C$ESCAPE
024604 000014                                   .WORD   10000$-.
024606 005202                                   INC     R2              ;NEXT DATA
024610 005205                                   INC     R5              ;NEXT ADDRESS
024612 022705 000010                            CMP     #10,R5          ;DONE YET?
024616 001324                                   BNE     1$              ;BR IF NO
024620                                          ENDSEG
024620                                  10000$:
024620 104405                                   TRAP    C$ESEG
024622                                          EXIT    TST
024622 104432                                   TRAP    C$EXIT
024624 000012                                   .WORD   L10127-.
024626    000    377    000            5$:     .BYTE   0,-1,0,-1,125,252,125,252
024631    377    125    252
024634    125    252
```

```
                                        .EVEN
024636                                  ENDTST
024636                                  L10127:
024636  104401                                  TRAP    C$ETST

024640                                          BADHEAD
                                                ;******************** TEST 46 ********************
                                                ;*ALU TEST
                                                ;*TEST OF ALU FUNCTION SEL A WITH C BIT CLEARED
                                                ;*ALU FUNCTION (A)    CODE=10
                                                ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
024640                                          BADHEAD
                                                ;******************** TEST 46 ********************

024640                                  BGNTST
024640                                  T46::
024640                                          MYINT
024640  013701  002716                          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
024644                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
024644  004537  003156                          JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
024650  005005                                  CLR     R5              ;MEM + SP ADDRESS
024652  012702  025032                          MOV     #5$,R2          ;POINTER TO CORRECT DATA
024656  004737  003640                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
024662  002654                                  MEMDAT                  ;POINTER TO DATA
024664  004737  004012                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
024670  002664                                  SPDAT                   ;POINTER TO DATA
024672                                          BGNSEG
024672  104404                                  TRAP    C$BSEG
024674  004737  004060          1$:             JSR     PC,CLRC         ;CLEAR C BIT!
024700  042737  000017  024716                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
024706  050537  024716                          BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
024712                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024712  004537  003244                          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
024716  010000                  2$:             010000                  ;LOAD MAR
024720  042737  000017  024736                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
024726  050537  024736                          BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
024732                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024732  004537  003244                          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
024736  040600                  3$:             040400!<10*20>          ;BR   SEL A
024740                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024740  004537  003244                          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
024744  061224                                  61224                   ;MOVE BR TO PORT4
024746  111237  002636                          MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
024752  116104  000004                          MOVB    4(R1),R4        ;PUT "FOUND" IN R4
024756  123704  002636                          CMPB    $GDDAT,R4               ;DATA CORRECT?
024762  001411                                  BEQ     4$              ;BR IF YES
024764                                          ERROR   15,YES          ;ALU ERROR
024776  104455                                  TRAP    C$ERDF
025000  000017                                  .WORD   15
025002  004754                                  .WORD   EM15
025004  007120                                  .WORD   ERR15
025006                          4$:             ESCAPE  SEG
025006  104410                                  TRAP    C$ESCAPE
025010  000014                                  .WORD   10000$-.
```

```
6269  025012  005202                          INC     R2              ;NEXT DATA
6270  025014  005205                          INC     R5              ;NEXT DATA
6271  025016  022705  000010                 CMP     #10,R5          ;DONE YET?
6272  025022  001324                          BNE     1$              ;BR IF NO
6273  025024                                  ENDSEG
6274  025024                          10000$:
6275  025024  104405                          TRAP    C$ESEG
6276  025026                                  EXIT    TST
6277  025026  104432                          TRAP    C$EXIT
6278  025030  000012                          .WORD   L10130-.
6279  025032     000     000     377  5$:     .BYTE   0,0,-1,-1,125,125,252,252
6280  025035     377     125     125
6281  025040     252     252
6282
6283                                  .EVEN
6284  025042                          ENDTST
6285  025042                          L10130:
6286  025042  104401                          TRAP    C$ETST
6287
6288  025044                                  BADHEAD
6289                                          ;******************** TEST 47 ********************
6290                                          ;*ALU TEST
6291                                          ;*TEST OF ALU FUNCTION A OR NOTB WITH C BIT CLEARED
6292                                          ;*ALU FUNCTION (A OR NOTB)    CODE=12
6293                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6294                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
6295  025044                                  BADHEAD
6296                                          ;******************** TEST 47 ********************
6297
6298  025044                          BGNTST
6299  025044                          T47::
6300  025044                                  MYINT
6301  025044  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6302  025050                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
6303  025050  004537  003156                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
6304  025054  005005                          CLR     R5              ;MEM + SP ADDRESS
6305  025056  012702  025236                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
6306  025062  004737  003640                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
6307  025066  002654                          MEMDAT                  ;POINTER TO DATA
6308  025070  004737  004012                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
6309  025074  002664                          SPDAT                   ;POINTER TO DATA
6310  025076                                  BGNSEG
6311  025076  104404                          TRAP    C$BSEG
6312  025100  004737  004060          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
6313  025104  042737  000017  025122          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
6314  025112  050537  025122                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
6315  025116                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6316  025116  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6317  025122  010000                  2$:     010000                  ;LOAD MAR
6318  025124  042737  000017  025142          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
6319  025132  050537  025142                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
6320  025136                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6321  025136  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6322  025142  040640                  3$:     040400!<12*20>          ;BR   A OR NOTB
6323  025144                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6324  025144  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
```

```
6325  025150  061224                          61224                   ;MOVE BR TO PORT4
6326  025152  111237  002636                  MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
6327  025156  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
6328  025162  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
6329  025166  001411                          BEQ     4$              ;BR IF YES
6330  025170                                  ERROR   15,YES          ;ALU ERROR
6331  025202  104455                          TRAP    C$ERDF
6332  025204  000017                          .WORD   15
6333  025206  004754                          .WORD   EM15
6334  025210  007120                          .WORD   ERR15
6335  025212                          4$:     ESCAPE  SEG
6336  025212  104410                          TRAP    C$ESCAPE
6337  025214  000014                          .WORD   10000$-.
6338  025216  005202                          INC     R2              ;NEXT DATA
6339  025220  005205                          INC     R5              ;NEXT DATA
6340  025222  022705  000010                  CMP     #10,R5          ;DONE YET?
6341  025226  001324                          BNE     1$              ;BR IF NO
6342  025230                                  ENDSEG
6343  025230                          10000$:
6344  025230  104405                          TRAP    C$ESEG
6345  025232                                  EXIT    TST
6346  025232  104432                          TRAP    C$EXIT
6347  025234  000012                          .WORD   L10131-.
6348  025236     377     000     377  5$:     .BYTE   -1,0,-1,-1,-1,125,252,-1
6349  025241     377     377     125
6350  025244     252     377
6351
6352                                  .EVEN
6353  025246                          ENDTST
6354  025246                          L10131:
6355  025246  104401                          TRAP    C$ETST
6356
6357  025250                                  BADHEAD
6358                                          ;******************** TEST 48 ********************
6359                                          ;*ALU TEST
6360                                          ;*TEST OF ALU FUNCTION A AND B WITH C BIT CLEARED
6361                                          ;*ALU FUNCTION (A AND B)    CODE=13
6362                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6363                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
6364  025250                                  BADHEAD
6365                                          ;******************** TEST 48 ********************
6366
6367  025250                          BGNTST
6368  025250                          T48::
6369  025250                                  MYINT
6370  025250  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6371  025254                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
6372  025254  004537  003156                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
6373  025260  005005                          CLR     R5              ;MEM + SP ADDRESS
6374  025262  012702  025442                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
6375  025266  004737  003640                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
6376  025272  002654                          MEMDAT                  ;POINTER TO DATA
6377  025274  004737  004012                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
6378  025300  002664                          SPDAT                   ;POINTER TO DATA
6379  025302                                  BGNSEG
6380  025302  104404                          TRAP    C$BSEG
```

```
6381  025304  004737  004060          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
6382  025310  042737  000017  025326          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
6383  025316  050537  025326                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
6384  025322                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6385  025322  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6386  025326  010000                  2$:     010000                  ;LOAD MAR
6387  025330  042737  000017  025346          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
6388  025336  050537  025346                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
6389  025342                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6390  025342  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6391  025346  040660                  3$:     040400!<13*20>          ;BR   A AND B
6392  025350                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6393  025350  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6394  025354  061224                          61224                   ;MOVE BR TO PORT4
6395  025356  111237  002636                  MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
6396  025362  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
6397  025366  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
6398  025372  001411                          BEQ     4$              ;BR IF YES
6399  025374                                  ERROR   15,YES          ;ALU ERROR
6400  025406  104455                          TRAP    C$ERDF
6401  025410  000017                          .WORD   15
6402  025412  004754                          .WORD   EM15
6403  025414  007120                          .WORD   ERR15
6404  025416                          4$:     ESCAPE  SEG
6405  025416  104410                          TRAP    C$ESCAPE
6406  025420  000014                          .WORD   10000$-.
6407  025422  005202                          INC     R2              ;NEXT DATA
6408  025424  005205                          INC     R5              ;NEXT DATA
6409  025426  022705  000010                  CMP     #10,R5          ;DONE YET?
6410  025432  001324                          BNE     1$              ;BR IF NO
6411  025434                                  ENDSEG
6412  025434                          10000$:
6413  025434  104405                          TRAP    C$ESEG
6414  025436                                  EXIT    TST
6415  025436  104432                          TRAP    C$EXIT
6416  025440  000012                          .WORD   L10132-.
6417  025442     000     000     000  5$:     .BYTE   0,0,0,-1,125,0,0,252
6418  025445     377     125     000
6419  025450     000     252
6420
6421                                  .EVEN
6422  025452                          ENDTST
6423  025452                          L10132:
6424  025452  104401                          TRAP    C$ETST
6425
6426  025454                                  BADHEAD
6427                                          ;******************** TEST 49 ********************
6428                                          ;*ALU TEST
6429                                          ;*TEST OF ALU FUNCTION A OR B WITH C BIT CLEARED
6430                                          ;*ALU FUNCTION (A OR B)    CODE=14
6431                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6432                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
6433  025454                                  BADHEAD
6434                                          ;******************** TEST 49 ********************
6435
6436  025454                          BGNTST
```

```
6437  025454                          T49::
6438  025454:
6439  025454  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6440  025460                          MSTCLR                          ;MASTER CLEAR M8200,4,7
6441  025460  004537  003156                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
6442  025464  005005                          CLR     R5              ;MEM + SP ADDRESS
6443  025466  012702  025646                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
6444  025472  004737  003640                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
6445  025476  002654                          MEMDAT                  ;POINTER TO DATA
6446  025500  004737  004012                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
6447  025504  002664                          SPDAT                   ;POINTER TO DATA
6448  025506                          BGNSEG
6449  025506  104404                          TRAP    C$BSEG
6450  025510  004737  004060          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
6451  025514  042737  000017  025532          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
6452  025522  050537  025532                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
6453  025526                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6454  025526  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6455  025532  010000                  2$:     010000                  ;LOAD MAR
6456  025534  042737  000017  025552          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
6457  025542  050537  025552                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
6458  025546                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6459  025546  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6460  025552  040700                  3$:     040400!<14*20>          ;BR   A OR B
6461  025554                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6462  025554  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6463  025560  061224                          61224                   ;MOVE BR TO PORT4
6464  025562  111237  002636                  MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
6465  025566  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
6466  025572  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
6467  025576  001411                          BEQ     4$              ;BR IF YES
6468  025600                          .       ERROR   15,YES          ;ALU ERROR
6469  025612  104455                          TRAP    C$ERDF
6470  025614  000017                          .WORD   15
6471  025616  004754                          .WORD   EM15
6472  025620  007120                          .WORD   ERR15
6473  025622                          4$:     ESCAPE  SEG
6474  025622  104410                          TRAP    C$ESCAPE
6475  025624  000014                          .WORD   10000$-.
6476  025626  005202                          INC     R2              ;NEXT DATA
6477  025630  005205                          INC     R5              ;NEXT DATA
6478  025632  022705  000010                  CMP     #10,R5          ;DONE YET?
6479  025636  001324                          BNE     1$              ;BR IF NO
6480  025640                          ENDSEG
6481  025640                          10000$:
6482  025640  104405                          TRAP    C$ESEG
6483  025642                          EXIT    TST
6484  025642  104432                          TRAP    C$EXIT
6485  025644  000012                          .WORD   L10133-.
6486  025646     000     377     377  5$:     .BYTE   0,-1,-1,-1,125,-1,-1,252
6487  025651     377     125     377
6488  025654     377     252
6489
6490                                  .EVEN
6491  025656                          ENDTST
6492  025656                          L10133:
```

```
6493  025656  104401                          TRAP    C$ETST
6494
6495  025660                                  BADHEAD
6496                                          ;******************** TEST 50 ********************
6497                                          ;*ALU TEST
6498                                          ;*TEST OF ALU FUNCTION A XOR B WITH C BIT CLEARED
6499                                          ;*ALU FUNCTION (A XOR B)    CODE=15
6500                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6501                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
6502  025660                                  BADHEAD
6503                                          ;******************** TEST 50 ********************
6504
6505  025660                          BGNTST
6506  025660                          T50::
6507  025660                                  MYINIT
6508  025660  013701  002716                 MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6509  025664                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
6510  025664  004537  003156                 JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
6511  025670  005005                         CLR     R5              ;MEM + SP ADDRESS
6512  025672  012702  026052                 MOV     #5$,R2          ;POINTER TO CORRECT DATA
6513  025676  004737  003640                 JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
6514  025702  002654                         MEMDAT                  ;POINTER TO DATA
6515  025704  004737  004012                 JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
6516  025710  002664                         SPDAT                   ;POINTER TO DATA
6517  025712                                  BGNSEG
6518  025712  104404                         TRAP    C$BSEG
6519  025714  004737  004060          1$:    JSR     PC,CLRC         ;CLEAR C BIT!
6520  025720  042737  000017  025736         BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
6521  025726  050537  025736                 BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
6522  025732                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6523  025732  004537  003244                 JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6524  025736  010000                  2$:    010000                  ;LOAD MAR
6525  025740  042737  000017  025756         BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
6526  025746  050537  025756                 BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
6527  025752                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6528  025752  004537  003244                 JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6529  025756  040720                  3$:    040400!<15*20>          ;BR   A XOR B
6530  025760                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6531  025760  004537  003244                 JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6532  025764  061224                         61224                   ;MOVE BR TO PORT4
6533  025766  111237  002636                 MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
6534  025772  116104  000004                 MOVB    4(R1),R4        ;PUT "FOUND" IN R4
6535  025776  123704  002636                 CMPB    $GDDAT,R4               ;DATA CORRECT?
6536  026002  001411                         BEQ     4$              ;BR IF YES
6537  026004                                  ERROR   15,YES          ;ALU ERROR
6538  026016  104455                         TRAP    C$ERDF
6539  026020  000017                         .WORD   15
6540  026022  004754                         .WORD   EM15
6541  026024  007120                         .WORD   ERR15
6542  026026                          4$:    ESCAPE  SEG
6543  026026  104410                         TRAP    C$ESCAPE
6544  026030  000014                         .WORD   10000$-.
6545  026032  005202                         INC     R2              ;NEXT DATA
6546  026034  005205                         INC     R5              ;NEXT DATA
6547  026036  022705  000010                 CMP     #10,R5          ;DONE YET?
6548  026042  001324                         BNE     1$              ;BR IF NO
```

```
 6549  026044                                    ENDSEG
 6550  026044                              10000$:
 6551  026044  104405                            TRAP    C$ESEG
 6552  026046                                    EXIT    TST
 6553  026046  104432                            TRAP    C$EXIT
 6554  026050  000012                            .WORD   L10134-.
 6555  026052     000     377     377  5$:      .BYTE   0,-1,-1,0,0,-1,-1,0
 6556  026055     000     000     377
 6557  026060     377     000
 6558
 6559                                    .EVEN
 6560  026062                            ENDTST
 6561  026062                            L10134:
 6562  026062  104401                            TRAP    C$ETST
 6563
 6564  026064                                    BADHEAD
 6565                                            ;******************** TEST 51 ********************
 6566                                            ;*ALU TEST
 6567                                            ;*TEST OF ALU FUNCTION ADD WITH C BIT CLEARED
 6568                                            ;*ALU FUNCTION (A PLUS B)    CODE=00
 6569                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 6570                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 6571  026064                                    BADHEAD
 6572                                            ;******************** TEST 51 ********************
 6573
 6574  026064                            BGNTST
 6575  026064                            T51::
 6576  026064                                    MYINT
 6577  026064  013701  002716                    MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6578  026070                                    MSTCLR                  ;MASTER CLEAR M8200,4,7
 6579  026070  004537  003156                    JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 6580  026074  005005                            CLR     R5              ;MEM + SP ADDRESS
 6581  026076  012702  026256                    MOV     #5$,R2          ;POINTER TO CORRECT DATA
 6582  026102  004737  003640                    JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 6583  026106  002654                            MEMDAT                  ;POINTER TO DATA
 6584  026110  004737  004012                    JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 6585  026114  002664                            SPDAT                   ;POINTER TO DATA
 6586  026116                                    BGNSEG
 6587  026116  104404                            TRAP    C$BSEG
 6588  026120  004737  004060            1$:     JSR     PC,CLRC         ;CLEAR C BIT!
 6589  026124  042737  000017  026142            BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 6590  026132  050537  026142                    BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 6591  026136                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 6592  026136  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6593  026142  010000                    2$:     010000                  ;LOAD MAR
 6594  026144  042737  000017  026162            BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 6595  026152  050537  026162                    BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 6596  026156                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 6597  026156  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6598  026162  040400                    3$:     040400!<00*20>          ;BR   ADD
 6599  026164                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 6600  026164  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6601  026170  061224                            61224                   ;MOVE BR TO PORT4
 6602  026172  111237  002636                    MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
 6603  026176  116104  000004                    MOVB    4(R1),R4        ;PUT "FOUND" IN R4
 6604  026202  123704  002636                    CMPB    $GDDAT,R4               ;DATA CORRECT?
```

```
6605  026206  001411                          BEQ     4$                ;BR IF YES
6606  026210                                  ERROR   15,YES            ;ALU ERROR
6607  026222  104455                          TRAP    C$ERDF
6608  026224  000017                          .WORD   15
6609  026226  004754                          .WORD   EM15
6610  026230  007120                          .WORD   ERR15
6611  026232                          4$:     ESCAPE  SEG
6612  026232  104410                          TRAP    C$ESCAPE
6613  026234  000014                          .WORD   10000$-.
6614  026236  005202                          INC     R2                ;NEXT DATA
6615  026240  005205                          INC     R5                ;NEXT DATA
6616  026242  022705  000010                  CMP     #10,R5            ;DONE YET?
6617  026246  001324                          BNE     1$                ;BR IF NO
6618  026250                                  ENDSEG
6619  026250                          10000$:
6620  026250  104405                          TRAP    C$ESEG
6621  026252                                  EXIT    TST
6622  026252  104432                          TRAP    C$EXIT
6623  026254  000012                          .WORD   L10135-.
6624  026256     000     377     377  5$:     .BYTE   0,-1,-1,376,252,-1,-1,124
6625  026261     376     252     377
6626  026264     377     124
6627
6628                                  .EVEN
6629  026266                          ENDTST
6630  026266                          L10135:
6631  026266  104401                          TRAP    C$ETST
6632
6633  026270                                  BADHEAD
6634                                  ;******************** TEST 52 ********************
6635                                  ;*ALU TEST
6636                                  ;*TEST OF ALU FUNCTION 2A W.C WITH C BIT CLEARED
6637                                  ;*ALU FUNCTION (A PLUS A PLUS C)        CODE=6
6638                                  ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6639                                  ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
6640  026270                          BADHEAD
6641                                  ;******************** TEST 52 ********************
6642
6643  026270                          BGNTST
6644  026270                          T52::
6645  026270                                  MYINT
6646  026270  013701  002716                  MOV     KMCSR,R1          ;GET DEVICE ADDRESS.
6647  026274                                  MSTCLR                    ;MASTER CLEAR M8200,4,7
6648  026274  004537  003156                  JSR     R5,..MSTCLR               ;CLEAR M8200,4,7
6649  026300  005005                          CLR     R5                ;MEM + SP ADDRESS
6650  026302  012702  026462                  MOV     #5$,R2            ;POINTER TO CORRECT DATA
6651  026306  004737  003640                  JSR     PC,MEMLD          ;LOAD 8 WORDS OF MAIN MEMORY
6652  026312  002654                          MEMDAT                    ;POINTER TO DATA
6653  026314  004737  004012                  JSR     PC,SPLD           ;LOAD 8 WORDS OF SP
6654  026320  002664                          SPDAT                     ;POINTER TO DATA
6655  026322                                  BGNSEG
6656  026322  104404                          TRAP    C$BSEG
6657  026324  004737  004060          1$:     JSR     PC,CLRC           ;CLEAR C BIT!
6658  026330  042737  000017  026346          BIC     #17,2$            ;CLEAR ADDRESS FIELD OF INSTRUCTION
6659  026336  050537  026346                  BIS     R5,2$             ;ADD ADDRESS TO INSTRUCTION
6660  026342                                  ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
6661  026342  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6662  026346  010000                2$:     010000                          ;LOAD MAR
6663  026350  042737  000017  026366        BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
6664  026356  050537  026366                BIS     R5,3$                   ;ADD ADDRESS TO INSTRUCTION
6665  026362                                ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6666  026362  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6667  026366  040540                3$:     040400!<6*20>                   ;BR   2A W/C
6668  026370                                ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6669  026370  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6670  026374  061224                        61224                           ;MOVE BSR TO PORT4
6671  026376  111237  002636                MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
6672  026402  116104  000004                MOVB    4(R1),R4                ;PUT "FOUND" IN R4
6673  026406  123704  002636                CMPB    $GDDAT,R4               ;DATA CORRECT?
6674  026412  001411                        BEQ     4$                      ;BR IF YES
6675  026414                                ERROR   15,YES                  ;ALU ERROR
6676  026426  104455                        TRAP    C$ERDF
6677  026430  000017                        .WORD   15
6678  026432  004754                        .WORD   EM15
6679  026434  007120                        .WORD   ERR15
6680  026436                        4$:     ESCAPE  SEG
6681  026436  104410                        TRAP    C$ESCAPE
6682  026440  000014                        .WORD   10000$-.
6683  026442  005202                        INC     R2                      ;NEXT DATA
6684  026444  005205                        INC     R5                      ;NEXT ADDRESS
6685  026446  022705  000010                CMP     #10,R5                  ;DONE YET?
6686  026452  001324                        BNE     1$                      ;BR IF NO
6687  026454                                ENDSEG
6688  026454                        10000$:
6689  026454  104405                        TRAP    C$ESEG
6690  026456                                EXIT    TST
6691  026456  104432                        TRAP    C$EXIT
6692  026460  000012                        .WORD   L10136-.
6693  026462     000     000     376  5$:     .BYTE   0,0,376,376,252,252,124,124
6694  026465     376     252     252
6695  026470     124     124
6696
6697                                .EVEN
6698  026472                        ENDTST
6699  026472                        L10136:
6700  026472  104401                        TRAP    C$ETST
6701
6702  026474                                BADHEAD
6703                                        ;******************** TEST 53 ********************
6704                                        ;*ALU TEST
6705                                        ;*TEST OF ALU FUNCTION SUB WITH C BIT CLEARED
6706                                        ;*ALU FUNCTION (A-B)    CODE=16
6707                                        ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6708                                        ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
6709  026474                                BADHEAD
6710                                        ;******************** TEST 53 ********************
6711
6712  026474                        BGNTST
6713  026474                        T53::
6714  026474                                MYINT
6715  026474  013701  002716                MOV     KMCSR,R1                ;GET DEVICE ADDRESS.
6716  026500                                MSTCLR                          ;MASTER CLEAR M8200,4,7
```

```
6717  026500  004537  002156                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
6718  026504  005005                          CLR     R5                      ;MEM + SP ADDRESS
6719  026506  012702  026670                  MOV     #5$,R2                  ;POINTER TO CORRECT DATA
6720  026512  004737  003640                  JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
6721  026516  002654                          MEMDAT                          ;POINTER TO DATA
6722  026520  004737  004012                  JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
6723  026524  002664                          SPDAT                           ;POINTER TO DATA
6724  026526                                  BGNSEG
6725  026526  104404                          TRAP    C$BSEG
6726  026530  004737  004060          1$:     JSR     PC,CLRC                 ;CLEAR C BIT!
6727  026534  042737  000017  026552          BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
6728  026542  050537  026552                  BIS     R5,2$                   ;ADD ADDRESS TO INSTRUCTION
6729  026546                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6730  026546  004537  003244                  JSR     R5,..ROMCLK                     ;CLOCK INSTRUCTION
6731  026552  010000                  2$:     010000                          ;LOAD MAR
6732  026554  042737  000017  026572          BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
6733  026562  050537  026572                  BIS     R5,3$                   ;ADD ADDRESS TO INSTRUCTION
6734  026566                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6735  026566  004537  003244                  JSR     R5,..ROMCLK                     ;CLOCK INSTRUCTION
6736  026572  040740                  3$:     040400!<16*20>                  ;BR _ SUB
6737  026574                                  ROMCLK
6738  026574  004537  003244                  JSR     R5,..ROMCLK                     ;CLOCK INSTRUCTION
6739  026600  061224                          61224                           ;MOVE BR TO PORT4
6740  026602  111237  002636                  MOVB    (R2),$GDDAT                     ;PUT "EXPECTED" IN $GDDAT
6741  026606  116104  000004                  MOVB    4(R1),R4                ;PUT "FOUND" IN R4
6742  026612  123737  002636  002636          CMPB    $GDDAT,$GDDAT                   ;DATA CORRECT?
6743  026620  001411                          BEQ     4$                      ;BR IF YES
6744  026622                                  ERROR   15,YES                  ;ALU ERROR
6745  026634  104455                          TRAP    C$ERDF
6746  026636  000017                          .WORD   15
6747  026640  004754                          .WORD   EM15
6748  026642  007120                          .WORD   ERR15
6749  026644                          4$:     ESCAPE  SEG
6750  026644  104410                          TRAP    C$ESCAPE
6751  026646  000014                          .WORD   10000$-.
6752  026650  005202                          INC     R2                      ;NEXT DATA
6753  026652  005205                          INC     R5                      ;NEXT ADDRESS
6754  026654  022705  000010                  CMP     #10,R5                  ;DONE YET?
6755  026660  001323                          BNE     1$                      ;BR IF NO
6756  026662                                  ENDSEG
6757  026662                          10000$:
6758  026662  104405                          TRAP    C$ESEG
6759  026664                                  EXIT    TST
6760  026664  104432                          TRAP    C$EXIT
6761  026666  000012                          .WORD   L10137-.
6762  026670     000     001     377  5$:     .BYTE   0,1,-1,0,0,253,125,0
6763  026673     000     000     253
6764  026676     125     000
6765
6766
6767                                  .EVEN
6768  026700                          ENDTST
6769  026700                          L10137:
6770  026700  104401                          TRAP    C$ETST
6771
6772
```

```
 6773  026702                                  BADHEAD
 6774                                          ;********************* TEST 54 *********************
 6775                                          ;*ALU TEST
 6776                                          ;*TEST OF ALU FUNCTION ADD W/C WITH C BIT CLEARED
 6777                                          ;*ALU FUNCTION (A PLUS B PLUS C)          CODE=01
 6778                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 6779                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 6780  026702                                  BADHEAD
 6781                                          ;********************* TEST 54 *********************
 6782
 6783  026702                          BGNTST
 6784  026702                          T54::
 6785  026702                                  MYINT
 6786  026702  013701  002716                 MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6787  026706                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 6788  026706  004537  003156                 JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 6789  026712  005005                         CLR     R5              ;MEM + SP ADDRESS
 6790  026714  012702  027074                 MOV     #5$,R2          ;POINTER TO CORRECT DATA
 6791  026720  004737  003640                 JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 6792  026724  002654                         MEMDAT                  ;POINTER TO DATA
 6793  026726  004737  004012                 JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 6794  026732  002664                         SPDAT                   ;POINTER TO DATA
 6795  026734                                  BGNSEG
 6796  026734  104404                         TRAP    C$BSEG
 6797  026736  004737  004060         1$:     JSR     PC,CLRC         ;CLEAR C BIT!
 6798  026742  042737  000017  026760         BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 6799  026750  050537  026760                 BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 6800  026754                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 6801  026754  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6802  026760  010000                 2$:     010000                  ;LOAD MAR
 6803  026762  042737  000017  027000         BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 6804  026770  050537  027000                 BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 6805  026774                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 6806  026774  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6807  027000  040420                 3$:     040400!<01*20>          ;BR   ADD W/C
 6808  027002                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 6809  027002  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6810  027006  061224                         61224                   ;MOVE BR TO PORT4
 6811  027010  111237  002636                 MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
 6812  027014  116104  000004                 MOVB    4(R1),R4        ;PUT "FOUND" IN R4
 6813  027020  123704  002636                 CMPB    $GDDAT,R4               ;DATA CORRECT?
 6814  027024  001411                         BEQ     4$              ;BR IS YES
 6815  027026                                  ERROR   15,YES          ;ALU ERROR
 6816  027040  104455                         TRAP    C$ERDF
 6817  027042  000017                         .WORD   15
 6818  027044  004754                         .WORD   EM15
 6819  027046  007120                         .WORD   ERR15
 6820  027050                          4$:     ESCAPE  SEG
 6821  027050  104410                         TRAP    C$ESCAPE
 6822  027052  000014                         .WORD   10000$-.
 6823  027054  005202                         INC     R2              ;NEXT DATA
 6824  027056  005205                         INC     R5              ;NEXT ADDRESS
 6825  027060  022705  000010                 CMP     #10,R5          ;DONE YET?
 6826  027064  001324                         BNE     1$              ;BR IF NO
 6827  027066                                  ENDSEG
 6828  027066                          10000$:
```

```
6829  027066  104405                          TRAP    C$ESEG
6830  027070                                  EXIT    TST
6831  027070  104432                          TRAP    C$EXIT
6832  027072  000012                          .WORD   L10140-.
6833  027074    000     377     377   5$:     .BYTE   0,-1,-1,376,252,-1,-1,124
6834  027077    376     252     377
6835  027102    377     124
6836
6837                                  .EVEN
6838  027104                          ENDTST
6839  027104                          L10140:
6840  027104  104401                          TRAP    C$ETST
6841
6842
6843  027106                                  BADHEAD
6844                                          ;******************** TEST 55 ********************
6845                                          ;*ALU TEST
6846                                          ;*TEST OF ALU FUNCTION SUB W/C WITH C BIT CLEARED
6847                                          ;*ALU FUNCTION (A-B-C)  CODE=2
6848                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6849                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
6850  027106                                  BADHEAD
6851                                          ;******************** TEST 55 ********************
6852
6853  027106                          BGNTST
6854  027106                          T55::
6855  027106                                  MYINT
6856  027106  013701  002716                 MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
```

```
6857  027112                                    MSTCLR                  ;MASTER CLEAR M8200,4,7
6858  027112  004537  003156                    JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
6859  027116  005005                            CLR     R5              ;MEM + SP ADDRESS
6860  027120  012702  027300                    MOV     #5$,R2          ;POINTER TO CORRECT DATA
6861  027124  004737  003640                    JSR     PC, MEMLD       ;LOAD 8 WORDS OF MAIN MEMORY
6862  027130  002654                            MEMDAT                  ;POINTER TO DATA
6863  027132  004737  004012                    JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
6864  027136  002664                            SPDAT                   ;POINTER TO DATA
6865  027140                                    BGNSEG
6866  027140  104404                            TRAP    C$BSEG
6867  027142  004737  004060          1$:       JSR     PC,CLRC         ;CLEAR C BIT!
6868  027146  042737  000017  027164            BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
6869  027154  050537  027164                    BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
6870  027160                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6871  027160  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6872  027164  010000                  2$:       010000                  ;LOAD MAR
6873  027166  042737  000017  027204            BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
6874  027174  050537  027204                    BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
6875  027200                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6876  027200  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6877  027204  040440                  3$:       040400!<2*20>           ;BR _ SUB W/C
6878  027206                                    ROMCLK
6879  027206  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6880  027212  061224                            61224                   ;MOVE BR TO PORT4
6881  027214  111237  002636                    MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
6882  027220  116104  000004                    MOVB    4(R1),R4        ;PUT "FOUND" IN R4
6883  027224  123704  002636                    CMPB    $GDDAT,R4               ;DATA CORRECT?
6884  027230  001411                            BEQ     4$              ;BR IF YES
6885  027232                                    ERROR   15,YES          ;ALU ERROR
6886  027244  104455                            TRAP    C$ERDF
6887  027246  000017                            .WORD   15
6888  027250  004754                            .WORD   EM15
6889  027252  007120                            .WORD   ERR15
6890  027254                            4$:     ESCAPE  SEG
6891  027254  104410                            TRAP    C$ESCAPE
6892  027256  000014                            .WORD   10000$-.
6893  027260  005202                            INC     R2                      ;NEXT DATA
6894  027262  005205                            INC     R5                      ;NEXT ADDRESS
6895  027264  022705  000010                    CMP     #10,R5                  ;DONE YET?
6896  027270  001324                            BNE     1$                      ;BR IF NO
6897  027272                                    ENDSEG
6898  027272                            10000$:
6899  027272  104405                            TRAP    C$ESEG
6900  027274                                    EXIT    TST
6901  027274  104432                            TRAP    C$EXIT
6902  027276  000012                            .WORD   L10141-.
6903  027300     377     000     376    5$:     .BYTE   -1,0,376,-1,-1,252,124,-1
6904  027303     377     377     252
6905  027306     124     377
6906
6907
6908                                    .EVEN
6909  027310                            ENDTST
6910  027310                            L10141:
6911  027310  104401                            TRAP    C$ETST
6912
```

```
                                                BADHEAD
                                                ;******************** TEST 56 ********************
                                                ;*ALU TEST
                                                ;*TEST OF ALU FUNCTION INC A WITH C BIT CLEARED
                                                ;*ALU FUNCTION (A PLUS 1)        CODE=3
                                                ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
                                                BADHEAD
                                                ;******************** TEST 56 ********************

                                        BGNTST
                                        T56::
                                                MYINT
027312  013701  002716                          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
027316                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
027316  004537  003156                          JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
027322  012702  027504                          MOV     #5$,R2          ;POINTER TO CORRECT DATA
027326  005005                                  CLR     R5
027330  004737  003640                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMRY
027334  002654                                  MEMDAT                  ;POINTER TO DATA
027336  004737  004012                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
027342  002664                                  SPDAT                   ;POINTER TO DATA
027344                                          BGNSEG
027344  104404                                  TRAP    C$BSEG
027346  004737  004060          1$:             JSR     PC,CLRC         ;CLEAR C BIT!
027352  042737  000017  027370                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
027360  050537  027370                          BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
027364                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027364  004537  003244                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
027370  010000                  2$:             010000                  ;LOAD MAR
027372  042737  000017  027410                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
027400  050537  027410                          BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
027404                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027404  004537  003244                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
027410  040460                  3$:             040400!<3*20>           ;BR  INC A
027412                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027412  004537  003244                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
027416  061224                                  61224                   ;MOVE BR TO PORT4
027420  111237  002636                          MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
027424  116104  000004                          MOVB    4(R1),R4        ;PUT "FOUND" IN R4
027430  123704  002636                          CMPB    $GDDAT,R4               ;DATA CORRECT?
027434  001411                                  BEQ     4$              ;BR IF YES
027436                                          ERROR   15,YES          ;ALU ERROR
027450  104455                                  TRAP    C$ERDF
027452  000017                                  .WORD   15
027454  004754                                  .WORD   EM15
027456  007120                                  .WORD   ERR15
027460                          4$:             ESCAPE  SEG
027460  104410                                  TRAP    C$ESCAPE
027462  000014                                  .WORD   10000$-.
027464  005202                                  INC     R2              ;NEXT DATA
027466  005205                                  INC     R5
027470  022705  000010                          CMP     #10,R5          ;DONE YET?
027474  001324                                  BNE     1$              ;BR IF NO
027476                                          ENDSEG
```

```
 6969  027476                                  10000$:
 6970  027476  104405                                  TRAP    C$ESEG
 6971  027500                                          EXIT    TST
 6972  027500  104432                                  TRAP    C$EXIT
 6973  027502  000012                                  .WORD   L10142-.
 6974  027504     001      001      000  5$:          .BYTE   1,1,0,0,126,126,253,253
 6975  027507     000      126      126
 6976  027512     253      253
 6977
 6978                                          .EVEN
 6979  027514                                  ENDTST
 6980  027514                                  L10142:
 6981  027514  104401                                  TRAP    C$ETST
 6982
 6983
 6984  027516                                          BADHEAD
 6985                                                  ;******************** TEST 57 ********************
 6986                                                  ;*ALU TEST
 6987                                                  ;*TEST OF ALU FUNCTION 2A WITH C BIT CLEARED
 6988                                                  ;*ALU FUNCTION (A PLUS A)       CODE=5
 6989                                                  ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 6990                                                  ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 6991  027516                                          BADHEAD
 6992                                                  ;******************** TEST 57 ********************
 6993
 6994  027516                                  BGNTST
 6995  027516                                  T57::
 6996  027516                                          MYINT
 6997  027516  013701  002716                          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6998  027522                                          MSTCLR                  ;MASTER CLEAR DMC11
 6999  027522  004537  003156                          JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 7000  027526  005005                                  CLR     R5              ;MEM * SP ADDRESS
 7001  027530  012702  027710                          MOV     #5$,R2          ;POINTER TO CORRECT DATA
 7002  027534  004737  003640                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 7003  027540  002654                                  MEMDAT                  ;POINTER TO DATA
 7004  027542  004737  004012                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 7005  027546  002664                                  SPDAT                   ;POINTER TO DATA
 7006  027550                                          BGNSEG
 7007  027550  104404                                  TRAP    C$BSEG
 7008  027552  004737  004060          1$:             JSR     PC,CLRC         ;CLEAR C BIT!
 7009  027556  042737  000017  027574                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7010  027564  050537  027574                          BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 7011  027570                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7012  027570  004537  003244                          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7013  027574  010000                          2$:     010000                  ;LOAD MAR
 7014  027576  042737  000017  027614                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 7015  027604  050537  027614                          BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 7016  027610                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7017  027610  004537  003244                          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7018  027614  040520                          3$:     040400!<5*20>           ;BR   2A
 7019  027616                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7020  027616  004537  003244                          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7021  027622  061224                                  61224                   ;MOVE BR TO PORT4
 7022  027624  111237  002636                          MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
 7023  027630  116104  000004                          MOVB    4(R1),R4        ;PUT "FOUND" IN R4
 7024  027634  123704  002636                          CMPB    $GDDAT,R4               ;DATA CORRECT?
```

```
7025  027640  001411                          BEQ     4$              ;BR IF YES
7026  027642                                  ERROR   15,YES          ;ALU ERROR
7027  027654  104455                          TRAP    C$ERDF
7028  027656  000017                          .WORD   15
7029  027660  004754                          .WORD   EM15
7030  027662  007120                          .WORD   ERR15
7031  027664                          4$:     ESCAPE  SEG
7032  027664  104410                          TRAP    C$ESCAPE
7033  027666  000014                          .WORD   10000$-.
7034  027670  005202                          INC     R2              ;NEXT DATA
7035  027672  005205                          INC     R5              ;NEXT ADDRESS
7036  027674  022705  000010                  CMP     #10,R5          ;DONE YET?
7037  027700  001324                          BNE     1$              ;BR IF NO
7038  027702                                  ENDSEG
7039  027702                          10000$:
7040  027702  104405                          TRAP    C$ESEG
7041  027704                                  EXIT    TST
7042  027704  104432                          TRAP    C$EXIT
7043  027706  000012                          .WORD   L10143-.
7044  027710     000     000     376  5$:     .BYTE   0,0,376,376,252,252,124,124
7045  027713     376     252     252
7046  027716     124     124
7047
7048                                  .EVEN
7049  027720                          ENDTST
7050  027720                          L10143:
7051  027720  104401                          TRAP    C$ETST
7052
7053
7054  027722                                  BADHEAD
7055                                  ;********************* TEST 58 *********************
7056                                  ;*ALU TEST
7057                                  ;*TEST OF ALU FUNCTION A PLUS C WITH C BIT CLEARED
7058                                  ;*ALU FUNCTION (A PLUS C)       CODE=4
7059                                  ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7060                                  ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7061  027722                          BADHEAD
7062                                  ;********************* TEST 58 *********************
7063
7064  027722                  BGNTST
7065  027722                  T58::
7066  027722                          MYINT
7067  027722  013701  002716          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7068  027726                          MSTCLR                  ;MASTER CLEAR M8200,4,7
7069  027726  004537  003156          JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
7070  027732  005005                  CLR     R5              ;MEM + SP ADDRESS
7071  027734  012702  030114          MOV     #5$,R2          ;POINTER TO CORRECT DATA
7072  027740  004737  003640          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7073  027744  002654                  MEMDAT                  ;POINTER TO DATA
7074  027746  004737  004012          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7075  027752  002664                  SPDAT                   ;POINTER TO DATA
7076  027754                          BGNSEG
7077  027754  104404                  TRAP    C$BSEG
7078  027756  004737  004060  1$:     JSR     PC,CLRC         ;CLEAR C BIT!
7079  027762  042737  00C017  030000  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7080  027770  050537  030000          BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
```

```
7081  027774                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7082  027774  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7083  030000  010000                  2$:       010000                  ;LOAD MAR
7084  030002  042737  000017  030020            BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7085  030010  050537  030020                    BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
7086  030014                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=304
7087  030014  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7088  030020  040500                  3$:       040400!<4*20>           ;BR   A PLUS C
7089  030022                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCKL PC=5305
7090  030022  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7091  030026  061224                            61224                   ;MOVE BR TO PORT4
7092  030030  111237  002636                    MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
7093  030034  116104  000004                    MOVB    4(R1),R4        ;PUT "FOUND" IN R4
7094  030040  123704  002636                    CMPB    $GDDAT,R4               ;DATA CORRECT?
7095  030044  001411                            BEQ     4$              ;BR IS YES
7096  030046                                    ERROR   15,YES          ;ALU ERROR
7097  030060  104455                            TRAP    C$ERDF
7098  030062  000017                            .WORD   15
7099  030064  004754                            .WORD   EM15
7100  030066  007120                            .WORD   ERR15
7101  030070                            4$:     ESCAPE  SEG
7102  030070  104410                            TRAP    C$ESCAPE
7103  030072  000014                            .WORD   10000$-.
7104  030074  005202                            INC     R2              ;NEXT DATA
7105  030076  005205                            INC     R5              ;NEXT ADDRESS
7106  030100  022705  000010                    CMP     #10,R5          ;DONE YET?
7107  030104  001324                            BNE     1$              ;BR IF NO
7108  030106                                    ENDSEG
7109  030106                            10000$:
7110  030106  104405                            TRAP    C$ESEG
7111  030110                                    EXIT    TST
7112  030110  104432                            TRAP    C$EXIT
7113  030112  000012                            .WORD   L10144-.
7114  030114     000     000     377    5$:     .BYTE   0,0,-1,-1,125,125,252,252
7115  030117     377     125     125
7116  030122     252     252
7117
7118                                      .EVEN
7119  030124                              ENDTST
7120  030124                              L10144:
7121  030124  104401                            TRAP    C$ETST
7122
7123
7124  030126                                    BADHEAD
7125                                            ;******************** TEST 59 ********************
7126                                            ;*ALU TEST
7127                                            ;*TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT CLEARED
7128                                            ;*ALU FUNCTION (A-B-1)  CODE=17
7129                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7130                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7131  030126                                    BADHEAD
7132                                            ;******************** TEST 59 ********************
7133
7134  030126                              BGNTST
7135  030126                              T59::
7136  030126                                    MYINT
```

```
 7137  030126  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7138  030132                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 7139  030132  004537  003156                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 7140  030136  005005                          CLR     R5              ;MEM + SP ADDRESS
 7141  030140  012702  030320                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
 7142  030144  004737  003640                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 7143  030150  002654                          MEMDAT                  ;POINTER TO DATA
 7144  030152  004737  004012                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 7145  030156  002664                          SPDAT                   ;POINTER TO DATA
 7146  030160                                  BGNSEG
 7147  030160  104404                          TRAP    C$BSEG
 7148  030162  004737  004060          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
 7149  030166  042737  000017  030204          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7150  030174  050537  030204                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 7151  030200                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7152  030200  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7153  030204  010000                  2$:     010000                  ;LOAD MAR
 7154  030206  042737  000017  030224          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 7155  030214  050537  030224                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 7156  030220                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7157  030220  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7158  030224  040760                  3$:     040400!<17*20>          ;BR   2'S COMP SUB
 7159  030226                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7160  030226  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7161  030232  061224                          61224                   ;MOVE BR TO PORT4
 7162  030234  111237  002636                  MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
 7163  030240  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
 7164  030244  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
 7165  030250  001411                          BEQ     4$              ;BR IS YES
 7166  030252                                  ERROR   15,YES          ;ALU ERROR
 7167  030264  104455                          TRAP    C$ERDF
 7168  030266  000017                          .WORD   15
 7169  030270  004754                          .WORD   EM15
 7170  030272  007120                          .WORD   ERR15
 7171  030274                          4$:     ESCAPE  SEG
 7172  030274  104410                          TRAP    C$ESCAPE
 7173  030276  000014                          .WORD   10000$-.
 7174  030300  005202                          INC     R2                      ;NEXT DATA
 7175  030302  005205                          INC     R5                      ;NEXT ADDRESS
 7176  030304  022705  000010                  CMP     #10,R5                  ;DONE YET?
 7177  030310  001324                          BNE     1$                      ;BR IF NO
 7178  030312                                  ENDSEG
 7179  030312                          10000$:
 7180  030312  104405                          TRAP    C$ESEG
 7181  030314                                  EXIT    TST
 7182  030314  104432                          TRAP    C$EXIT
 7183  030316  000012                          .WORD   L10145-.
 7184  030320     377     000     376  5$:     .BYTE   -1,0,376,-1,-1,252,124,-1
 7185  030323     377     377     252
 7186  030326     124     377
 7187
 7188                                  .EVEN
 7189  030330                          ENDTST
 7190  030330                          L10145:
 7191  030330  104401                          TRAP    C$ETST
 7192
```

```
                                                BADHEAD
030332                                          ;******************** TEST 60 ********************
                                                ;*ALU TEST
                                                ;*TEST OF ALU FUNCTION DEC A WITH C BIT CLEARED
                                                ;*ALU FUNCTION (A-1)    CODE=7
                                                ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS

030332                                          BADHEAD
                                                ;******************** TEST 60 ********************

030332                                  BGNTST
030332                                  T60::
030332                                          MYINT
030332  013701  002716                          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
030336                                          MSTCLR                  ;MASTER CLEAR DMC11
030336  004537  003156                          JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
030342  005005                                  CLR     R5              ;MEM + SP ADDRESS
030344  012702  030524                          MOV     #5$,R2          ;POINTER TO CORRECT DATA
030350  004737  003640                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMMOR
030354  002654                                  MEMDAT                  ;POINTER TO DATA
030356  004737  004012                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
030362  002664                                  SPDAT                   ;POINTER TO DATA
030364                                          BGNSEG
030364  104404                                  TRAP    C$BSEG
030366  004737  004060          1$:             JSR     PC,CLRC         ;CLEAR C BIT!
030372  042737  000017  030410                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
030400  050537  030410                          BIS     R5,2$           ;ADD ADDRESS TO INSTRUCITON
030404                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030404  004537  003244                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
030410  010000                          2$:     010000                  ;LOAD MAR
030412  042737  000017  030430                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
030420  050537  030430                          BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
030424                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030424  004537  003244                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
030430  040560                          3$:     040400!<7*20>           ;BR   DEC A
030432                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030432  004537  003244                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
030436  061224                                  61224                   ;MOVE BR TO PORT4
030440  111237  002636                          MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
030444  116104  000004                          MOVB    4(R1),R4        ;PUT "FOUND" IN R4
030450  123704  002636                          CMPB    $GDDAT,R4               ;DATA CORRECT?
030454  001411                                  BEQ     4$              ;BR IF YES
030456                                          ERROR   15,YES          ;ALU ERROR
030470  104455                                  TRAP    C$ERDF
030472  000017                                  .WORD   15
030474  004754                                  .WORD   EM15
030476  007120                                  .WORD   ERR15
030500                                  4$:     ESCAPE  SEG
030500  104410                                  TRAP    C$ESCAPE
030502  000014                                  .WORD   10000$-.
030504  005202                                  INC     R2              ;NEXT DATA
030506  005205                                  INC      R5             ;NEXT ADDRESS
030510  022705  000010                          CMP     #10,R5          ;DONE YET?
030514  001324                                  BNE      1$             ;BR IF NO
```

```
 7249  030516                                    ENDSEG
 7250  030516                              10000$:
 7251  030516  104405                            TRAP    C$ESEG
 7252  030520                                    EXIT    TST
 7253  030520  104432                            TRAP    C$EXIT
 7254  030522  000012                            .WORD   L10146-.
 7255  030524     377     377     376  5$:      .BYTE   -1,-1,376,376,124,124,251,251
 7256  030527     376     124     124
 7257  030532     251     251
 7258
 7259                                    .EVEN
 7260  030534                            ENDTST
 7261  030534                            L10146:
 7262  030534  104401                            TRAP    C$ETST
 7263
 7264
 7265  030536                                    BADHEAD
 7266                                            ;******************** TEST 61 ********************
 7267                                            ;*ALU TEST
 7268                                            ;*TEST OF ALU FUNCTION SEL B WITH C BIT SET
 7269                                            ;*ALU FUNCTION (B)      CODE=11
 7270                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 7271                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 7272  030536                                    BADHEAD
 7273                                            ;******************** TEST 61 ********************
 7274
 7275  030536                            BGNTST
 7276  030536                            T61::
 7277  030536                                    MYINT
 7278  030536  013701  002716                    MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7279  030542                                    MSTCLR                  ;MASTER CLEAR M8200,4,7
 7280  030542  004537  003156                    JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 7281  030546  005005                            CLR     R5              ;MEM + SP ADDRESS
 7282  030550  012702  030730                    MOV     #5$,R2          ;POINTER TO CORRECT DATA
 7283  030554  004737  003640                    JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 7284  030560  002654                            MEMDAT                  ;POINTER TO DATA
 7285  030562  004737  004012                    JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 7286  030566  002664                            SPDAT                   ;POINTET TO DATA
 7287  030570                                    BGNSEG
 7288  030570  104404                            TRAP    C$BSEG
 7289  030572  004737  004076              1$:   JSR     PC,SETC         ;SET C BIT!
 7290  030576  042737  000017  030614            BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7291  030604  050537  030614                    BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 7292  030610                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7293  030610  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7294  030614  010000                      2$:   010000                  ;LOAD MAR
 7295  030616  042737  000017  030634            BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 7296  030624  050537  030634                    BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 7297  030630                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7298  030630  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7299  030634  040620                      3$:   040400!<11*20>          ;BR   SEL B
 7300  030636                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7301  030636  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7302  030642  061224                            61224                   ;MOVE BR TO PORT4
 7303  030644  111237  002636                    MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
 7304  030650  116104  000004                    MOVB    4(R1),R4        ;PUT "FOUND" IN R4
```

```
 7305  030654  123704  002636          CMPB    $GDDAT,R4               ;DATA CORRECT?
 7306  030660  001411                  BEQ     4$              ;BR IF YES
 7307  030662                          ERROR   23,YES          ;ALU ERROR
 7308  030674  104455                  TRAP    C$ERDF
 7309  030676  000027                  .WORD   23
 7310  030700  005226                  .WORD   EM23
 7311  030702  007404                  .WORD   ERR23
 7312  030704                  4$:     ESCAPE  SEG
 7313  030704  104410                  TRAP    C$ESCAPE
 7314  030706  000014                  .WORD   10000$-.
 7315  030710  005202                  INC     R2              ;NEXT DATA
 7316  030712  005205                  INC     R5              ;NEXT ADDRESS
 7317  030714  022705  000010          CMP     #10,R5          ;DONE YET?
 7318  030720  001324                  BNE     1$              ;BR IF NO
 7319  030722                          ENDSEG
 7320  030722                  10000$:
 7321  030722  104405                  TRAP    C$ESEG
 7322  030724                          EXIT    TST
 7323  030724  104432                  TRAP    C$EXIT
 7324  030726  000012                  .WORD   L10147-.
 7325  030730     000     377     000  5$:     .BYTE   0,-1,0,-1,125,252,125,252
 7326  030733     377     125     252
 7327  030736     125     252
 7328
 7329                                  .EVEN
 7330  030740                          ENDTST
 7331  030740                  L10147:
 7332  030740  104401                  TRAP    C$ETST
 7333
 7334
 7335  030742                          BADHEAD
 7336                                  ;******************** TEST 62 ********************
 7337                                  ;*ALU TEST
 7338                                  ;*TEST OF ALU FUNCTION SEL A WITH C BIT SET
 7339                                  ;*ALU FUNCTION (A)      CODE=10
 7340                                  ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 7341                                  ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 7342  030742                          BADHEAD
 7343                                  ;******************** TEST 62 ********************
```

```
7344
7345
7346  030742                                  BGNTST
7347  030742                                  T62::
7348  030742                                          MYINT
7349  030742  013701  002716                          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7350  030746                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
7351  030746  004537  003156                          JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
7352  030752  005005                                  CLR     R5              ;MEM + SP ADDRESS
7353  030754  012702  031134                          MOV     #5$,R2          ;POINTER TO CORRECT DATA
7354  030760  004737  003640                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7355  030764  002654                                  MEMDAT                  ;POINTER TO DATA
7356  030766  004737  004012                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7357  030772  002664                                  SPDAT                   ;POINTER TO DATA
7358  030774                                          BGNSEG
7359  030774  104404                                  TRAP    C$BSEG
7360  030776  004737  004076                  1$:     JSR     PC,SETC         ;SET C BIT!
7361  031002  042737  000017  031020                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7362  031010  050537  031020                          BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
7363  031014                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7364  031014  004537  003244                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7365  031020  010000                          2$:     010000                  ;LOAD MAR
7366  031022  042737  000017  031040                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCITON
7367  031030  050537  031040                          BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
7368  031034                                          ROMCLK                  ;NEXT WORD IS INSTRUCITON, ROMCLK PC=5304
7369  031034  004537  003244                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7370  031040  040600                          3$:     040400!<10*20>          ;BR   SEL A
7371  031042                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7372  031042  004537  003244                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7373  031046  061224                                  61224                   ;MOVE BR TO PORT4
7374  031050  111237  002636                          MOVB    (R2),$GDDAT             ;PUT 'EXPECTED" IN $GDDAT
7375  031054  116104  000004                          MOVB    4(R1),R4        ;PUT 'FOUND" IN R4
7376  031060  123704  002636                          CMPB    $GDDAT,R4               ;DATA CORRECT?
7377  031064  001411                                  BEQ     4$              ;BR IF YES
7378  031066                                          ERROR   23,YES          ;ALU ERROR
7379  031100  104455                                  TRAP    C$ERDF
7380  031102  000027                                  .WORD   23
7381  031104  005226                                  .WORD   EM23
7382  031106  007404                                  .WORD   ERR23
7383  031110                                  4$:     ESCAPE  SEG
7384  031110  104410                                  TRAP    C$ESCAPE
7385  031112  000014                                  .WORD   10000$-
7386  031114  005202                                  INC     R2              ;NEXT DATA
7387  031116  005205                                  INC     R5              ;NEXT ADDRESS
7388  031120  022705  000010                          CMP     #10,R5          ;DONE YET?
7389  031124  001324                                  BNE     1$              ;BR IF NO
7390  031126                                          ENDSEG
7391  031126                                  10000$:
7392  031126  104405                                  TRAP    C$ESEG
7393  031130                                          EXIT    TST
7394  031130  104432                                  TRAP    C$EXIT
7395  031132  000012                                  .WORD   L10150-.
7396  031134     000      000      377          5$:     .BYTE   0,0,-1,-1,125,125,252,252
7397  031137     377      125      125
7398  031142     252      252
7399
```

```
7400                                    .EVEN
7401  031144                            ENDTST
7402  031144                            L10150:
7403  031144  104401                            TRAP    C$ETST
7404
7405
7406  031146                                    BADHEAD
7407                                            ;******************** TEST 63 ********************
7408                                            ;*ALU TEST
7409                                            ;*TEST OF ALU FUNCTION A OR NOTB WITH C BIT SET
7410                                            ;*ALU FUNCTION (A OR NOTB)      CODE=12
7411                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7412                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7413  031146                                    BADHEAD
7414                                            ;******************** TEST 63 ********************
7415
7416  031146                            BGNTST
7417  031146                            T63::
7418  031146                                    MYINT
7419  031146  013701  002716                    MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7420  031152                                    MSTCLR                  ;MASTER CLEAR M8200,4,7
7421  031152  004537  003156                    JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
7422  031156  005005                            CLR     R5              ;MEM + SP ADDRESS
7423  031160  012702  031340                    MOV     #5$,R2          ;POINTER TO CORRECT DATA
7424  031164  004737  003640                    JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7425  031170  002654                            MEMDAT                  ;POINTER TO DATA
7426  031172  004737  004012                    JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7427  031176  002664                            SPDAT                   ;POINTER TO DATA
7428  031200                                    BGNSEG
7429  031200  104404                            TRAP    C$BSEG
7430  031202  004737  004076          1$:       JSR     PC,SETC         ;SET C BIT!
7431  031206  042737  000017  031224            BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7432  031214  050537  031224                    BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
7433  031220                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7434  031220  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7435  031224  010000                  2$:       010000                  ;LOAD MAR
7436  031226  042737  000017  031244            BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7437  031234  050537  031244                    BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
7438  031240                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7439  031240  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7440  031244  040640                  3$:       040400!<12*20>          ;BR   A OR NOTB
7441  031246                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7442  031246  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7443  031252  061224                            61224                   ;MOVE BR TO PORT4
7444  031254  111237  002636                    MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
7445  031260  116104  000004                    MOVB    4(R1),R4        ;PUT "FOUND" IN R4
7446  031264  123704  002636                    CMPB    $GDDAT,R4               ;DATA CORRECT?
7447  031270  001411                            BEQ     4$              ;BR IF YES
7448  031272                                    ERROR    15,YES         ;ALU ERROR
7449  031304  104455                            TRAP    C$ERDF
7450  031306  000017                            .WORD   15
7451  031310  004754                            .WORD   EM15
7452  031312  007120                            .WORD   ERR15
7453  031314                            4$:     ESCAPE  SEG
7454  031314  104410                            TRAP    C$ESCAPE
7455  031316  000014                            .WORD   10000$-.
```

```
7456  031320  005202                          INC     R2              ;NEXT DATA
7457  031322  005205                          INC     R5              ;NEXT ADDRESS
7458  031324  022705  000010                  CMP     #10,R5          ;DONE YET?
7459  031330  001324                          BNE     1$              ;BR IF NO
7460  031332                                  ENDSEG
7461  031332                          10000$:
7462  031332  104405                          TRAP    C$ESEG
7463  031334                                  EXIT    TST
7464  031334  104432                          TRAP    C$EXIT
7465  031336  000012                          .WORD   L10151-
7466  031340    377     000     377   5$:     .BYTE   -1,0,-1,-1,-1,125,252,-1
7467  031343    377     377     125
7468  031346    252     377
7469
7470                                  .EVEN
7471  031350                          ENDTST
7472  031350                          L10151:
7473  031350  104401                          TRAP    C$ETST
7474
7475
7476  031352                                  BADHEAD
7477                                          ;******************** TEST 64 ********.***********
7478                                          ;*ALU TEST
7479                                          ;*TEST OF ALU FUNCTION A AND B WITH C BIT SET
7480                                          ;ALU FUNCTION (A AND B) CODE=13
7481                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7482                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7483  031352                                  BADHEAD
7484                                          ;******************** TEST 64 ********************
7485
7486  031352                          BGNTST
7487  031352                          T64::
7488  031352                                  MYINT
7489  031352  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7490  031356                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
7491  031356  004537  003156                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
7492  031362  005005                          CLR     R5              ;MEM + SP ADDRESS
7493  031364  012702  031544                  MOV     #5$,R2          ;POINTER TO CORRECT ADDRESS
7494  031370  004737  003640                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7495  031374  002654                          MEMDAT                  ;POINTER TO DATA
7496  031376  004737  004012                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7497  031402  002664                          SPDAT                   ;POINTER TO DATA
7498  031404                                  BGNSEG
7499  031404  104404                          TRAP    C$BSEG
7500  031406  004737  004076          1$:     JSR     PC,SETC         ;SET C BIT!
7501  031412  042737  000017  031430          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7502  031420  050537  031430                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
7503  031424                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7504  031424  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7505  031430  010000                  2$:     010000                  ;LOAD MAR
7506  031432  042737  000017  031450          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7507  031440  050537  031450                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCITON
7508  031444                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7509  031444  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7510  031450  040660                  3$:     040400!<13*20>          ;BR   A AND B
7511  031452                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
 7512  031452  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7513  031456  061224                          61224                   ;MOVE BR TO PORT4
 7514  031460  111237  002636                  MOVB    (R2),$GDDAT             ;PUT 'EXPECTED''IN $GDDAT
 7515  031464  116104  000004                  MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
 7516  031470  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
 7517  031474  001411                          BEQ     4$              ;BR IF YES
 7518  031476                                  ERROR   23,YES          ;ALU ERROR
 7519  031510  104455                          TRAP    C$ERDF
 7520  031512  000027                          .WORD   23
 7521  031514  005226                          .WORD   EM23
 7522  031516  007404                          .WORD   ERR23
 7523  031520                          4$:     ESCAPE  SEG
 7524  031520  104410                          TRAP    C$ESCAPE
 7525  031522  000014                          .WORD   10000$-.
 7526  031524  005202                          INC     R2              ;NEXT DATA
 7527  031526  005205                          INC     R5              ;NEXT ADDRESS
 7528  031530  022705  000010                  CMP     #10,R5          ;DONE YET?
 7529  031534  001324                          BNE     1$              ;BR IF NO
 7530  031536                                  ENDSEG
 7531  031536                          10000$:
 7532  031536  104405                          TRAP    C$ESEG
 7533  031540                                  EXIT    TST
 7534  031540  104432                          TRAP    C$EXIT
 7535  031542  000012                          .WORD   L10152-.
 7536  031544     000     000     000  5$:     .BYTE   0,0,0,-1,125,0,0,252
 7537  031547     377     125     000
 7538  031552     000     252
 7539
 7540                                  .EVEN
 7541  031554                          ENDTST
 7542  031554                          L10152:
 7543  031554  104401                          TRAP    C$ETST
 7544
 7545
 7546  031556                                  BADHEAD
 7547                                          ;******************** TEST 65 ********************
 7548                                          ;*ALU TEST
 7549                                          ;*TEST OF ALU FUNCTION A OR B WITH C BIT SET
 7550                                          ;*ALU FUNCTION (A OR B) CODE=14
 7551                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 7552                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 7553  031556                                  BADHEAD
 7554                                          ;******************** TEST 65 ********************
 7555
 7556  031556                          BGNTST
 7557  031556                          T65::
 7558  031556                                  MYINT
 7559  031556  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7560  031562                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 7561  031562  004537  003156                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 7562  031566  005005                          CLR     R5              ;MEM + SP ADDRESS
 7563  031570  012702  031750                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
 7564  031574  004737  003640                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 7565  031600  002654                          MEMDAT                  ;POINTER TO DATA
 7566  031602  004737  004012                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 7567  031606  002664                          SPDAT                   ;POINTER TO DATA
```

```
7568  031610                                  BGNSEG
7569  031610  104404                          TRAP    C$BSEG
7570  031612  004737  004076          1$:     JSR     PC,SETC         ;SET C BIT!
7571  031616  042737  000017  031634          BIC #17,2$              ;CLEAR ADDRESS FIELD OF INSTRUCTION
7572  031624  050537  031634                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
7573  031630                                  ROMCLK                  ;NEXT WORD IS INSTRUCITON, ROMCLK PC=5304
7574  031630  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7575  031634  010000                  2$:     010000                  ;LOAD MAR
7576  031636  042737  000017  031654          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7577  031644  050537  031654                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
7578  031650                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7579  031650  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7580  031654  040700                  3$:     040400!<14*20>          ;BR   A OR B
7581  031656                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7582  031656  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7583  031662  061224                          61224                   ;MOVE BR TO PORT4
7584  031664  111237  002636                  MOVB    (R2),$GDDAT             ;PUT "EXPECTED"IN R4
7585  031670  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
7586  031674  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
7587  031700  001411                          BEQ     4$              ;BR IF YES
7588  031702                                  ERROR   23,YES          ;ALU ERROR
7589  031714  104455                          TRAP    C$ERDF
7590  031716  000027                          .WORD   23
7591  031720  005226                          .WORD   EM23
7592  031722  007404                          .WORD   ERR23
7593  031724                          4$:     ESCAPE  SEG
7594  031724  104410                          TRAP    C$ESCAPE
7595  031726  000014                          .WORD   10000$-.
7596  031730  005202                          INC     R2                      ;NEXT DATA
7597  031732  005205                          INC     R5                      ;NEXT ADDRESS
7598  031734  022705  000010                  CMP     #10,R5                  ;DONE YET?
7599  031740  001324                          BNE     1$                      ;BR IF NO
7600  031742                                  ENDSEG
7601  031742                          10000$:
7602  031742  104405                          TRAP    C$ESEG
7603  031744                                  EX'T    TST
7604  031744  104432                          TRAP    C$EXIT
7605  031746  000012                          .WORD   L10153-.
7606  031750     000     377     377  5$:     .BYTE   0,-1,-1,-1,125,-1,-1,252
7607  031753     377     125     377
7608  031756     377     252
7609
7610                                  .EVEN
7611  031760                          ENDTST
7612  031760                          L10153:
7613  031760  104401                          TRAP    C$ETST
7614
7615
7616  031762                                  BADHEAD
7617                                  ;******************** TEST 66 ********************
7618                                  ;*ALU TEST
7619                                  ;*TEST OF ALU FUNCTION A XOR B WITH C BIT SET
7620                                  ;*ALU FUNCTION (A XOR B)          CODE=15
7621                                  ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7622                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7623
```

```
 7624  031762                                  BADHEAD
 7625                                          ;******************** TEST 66 ********************
 7626  031762                          BGNTST
 7627  031762                          T66::
 7628  031762                                  MYINT
 7629  031762  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7630  031766                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 7631  031766  004537  003156                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 7632  031772  005005                          CLR     R5              ;MEM + SP ADDRESS
 7633  031774  012702  032154                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
 7634  032000  004737  003640                  JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
 7635  032004  002654                          MEMDAT                  ;POINTER TO DATA
 7636  032006  004737  004012                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 7637  032012  002664                          SPDAT                   ;POINTER TO DATA
 7638  032014                                  BGNSEG
 7639  032014  104404                          TRAP    C$BSEG
 7640  032016  004737  004076          1$:     JSR     PC,SETC         ;SET C BIT!
 7641  032022  042737  000017  032040          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7642  032030  050537  032040                  BIS     R5,2$           ;ADD ADDRESS TO INSTPUCTION
 7643  032034                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7644  032034  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7645  032040  010000                  2$:     010000                  ;LOAD MAR
 7646  032042  042737  000017  032060          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 7647  032050  050537  032060                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 7648  032054                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7649  032054  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7650  032060  040720                  3$:     040400!<15*20>          ;BR   A XOR B
 7651  032062                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7652  032062  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7653  032066  061224                          61224                   ;MOVE BR TO PORT4
 7654  032070  111237  002636                  MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
 7655  032074  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
 7656  032100  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
 7657  032104  001411                          BEQ     4$              ;BR IF YES
 7658  032106                                  ERROR   23,YES          ;ALU ERROR
 7659  032120  104455                          TRAP    C$ERDF
 7660  032122  000027                          .WORD   23
 7661  032124  005226                          .WORD   EM23
 7662  032126  007404                          .WORD   ERR23
 7663  032130                          4$:     ESCAPE  SEG
 7664  032130  104410                          TRAP    C$ESCAPE
 7665  032132  000014                          .WORD   10000$-.
 7666  032134  005202                          INC     R2              ;NEXT DATA
 7667  032136  005205                          INC     R5              ;NEXT ADDRESS
 7668  032140  022705  000010                  CMP     #10,R5          ;DONE YET?
 7669  032144  001324                          BNE     1$              ;BR IF NO
 7670  032146                                  ENDSEG
 7671  032146                          10000$:
 7672  032146  104405                          TRAP    C$ESEG
 7673  032150                                  EXIT    TST
 7674  032150  104432                          TRAP    C$EXIT
 7675  032152  000012                          .WORD   L10154-.
 7676  032154     000     377     377  5$:     .BYTE   0,-1,-1,0,0,-1,-1,0
 7677  032157     000     000     377
 7678  032162     377     000
 7679
```

```
7680                                    .EVEN
7681  032164                            ENDTST
7682  032164                            L10154:
7683  032164  104401                            TRAP    C$ETST
7684
7685
7686  032166                                    BADHEAD
7687                                      ;******************** TEST 67 ********************
7688                                      ;*ALU TEST
7689                                      ;*TEST OF ALU FUNCTION ADD WITH C BIT SET
7690                                      ;*ALU FUNCTION (A PLUS B)       CODE=00
7691                                      ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7692                                      ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7693  032166                              BADHEAD
7694                                      ;******************** TEST 67 ********************
7695
7696  032166                            BGNTST
7697  032166                            T67::
7698  032166                                    MYINT
7699  032166  013701  002716                    MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7700  032172                                    MSTCLR                  ;MASTER CLEAR M8200,4,7
7701  032172  004537  003156                    JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
7702  032176  005005                            CLR     R5              ;MEM + SP ADDRESS
7703  032200  012702  032360                    MOV     #5$,R2          ;POINTER TO CORRECT DATA
7704  032204  004737  003640                    JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7705  032210  002654                            MEMDAT                  ;POINTER TO DATA
7706  032212  004737  004012                    JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7707  032216  002664                            SPDAT                   ;OOINTER TO DATA
7708  032220                                    BGNSEG
7709  032220  104404                            TRAP    C$BSEG
7710  032222  004737  004076          1$:       JSR     PC,SETC         ;SET C BIT!
7711  032226  042737  000017  032244            BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7712  032234  050537  032244                    BIS     R5,2$           ;ADD ADDRESS TO INSTRUCITON
7713  032240                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7714  032240  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7715  032244  010000                  2$:       010000                  ;LOAD MAR
7716  032246  042737  000017  032264            BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7717  032254  050537  032264                    BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
7718  032260                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7719  032260  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7720  032264  040400                  3$:       040400!<00*20>          ;BR   ADD
7721  032266                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7722  032266  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7723  032272  061224                            61224                   ;MOVE BR TO PORT4
7724  032274  111237  002636                    MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
7725  032300  116104  000004                    MOVB    4(R1),R4        ;PUT "FOUND" IN R4
7726  032304  123704  002636                    CMPB    $GDDAT,R4               ;DATA CORRECT?
7727  032310  001411                            BEQ     4$              ;BR IF YES
7728  032312                                    ERROR   23,YES          ;ALU ERROR
7729  032324  104455                            TRAP    C$ERDF
7730  032326  000027                            .WORD   23
7731  032330  005226                            .WORD   EM23
7732  032332  007404                            .WORD   ERR23
7733  032334                            4$:     ESCAPE  SEG
7734  032334  104410                            TRAP    C$ESCAPE
7735  032336  000014                            .WORD   10000$-.
```

```
 7736  032340  005202                          INC     R2              ;NEXT DATA
 7737  032342  005205                          INC     R5              ;NEXT ADDRESS
 7738  032344  022705  000010                  CMP     #10,R5          ;DONE YET?
 7739  032350  001324                          BNE     1$              ;BR IF NO
 7740  032352                                  ENDSEG
 7741  032352                          10000$:
 7742  032352  104405                          TRAP    C$ESEG
 7743  032354                                  EXIT    TST
 7744  032354  104432                          TRAP    C$EXIT
 7745  032356  000012                          .WORD   L10155-.
 7746  032360     000     377     377  5$:     .BYTE   0,-1,-1,376,252,-1,-1,124
 7747  032363     376     252     377
 7748  032366     377     124
 7749
 7750                                  .EVEN
 7751  032370                          ENDTST
 7752  032370                          L10155:
 7753  032370  104401                          TRAP    C$ETST
 7754
 7755
 7756  032372                                  BADHEAD
 7757                                          ;******************** TEST 68 ********.***********
 7758                                          ;*ALU TEST
 7759                                          ;*TEST OF ALU FUNCTION 2A W/C WITH C BIT SET
 7760                                          ;*ALU FUNCTION (A PLUS A PLUS C)        CODE=6
 7761                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 7762                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 7763  032372                                  BADHEAD
 7764                                          ;******************** TEST 68 *********************
 7765
 7766  032372                          BGNTST
 7767  032372                          T68::
 7768  032372                                  MYINT
 7769  032372  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7770  032376                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 7771  032376  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 7772  032402  005005                          CLR     R5              ;MEM + SP ADDRESS
 7773  032404  012702  032564                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
 7774  032410  004737  003640                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 7775  032414  002654                          MEMDAT                  ;POINTER TO DATA
 7776  032416  004737  004012                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 7777  032422  002664                          SPDAT                   ;POINTER TO DAT
 7778  032424                                  BGNSEG
 7779  032424  104404                          TRAP    C$BSEG
 7780  032426  004737  004076          1$:     JSR     PC,SETC         ;SET C BIT!
 7781  032432  042737  000017  032450          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7782  032440  050537  032450                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 7783  032444                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7784  032444  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7785  032450  010000                  2$:     010000                  ;LOAD MAR
 7786  032452  042737  000017  032470          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 7787  032460  050537  032470                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 7788  032464                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=55304
 7789  032464  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7790  032470  040540                  3$:     040400!<6*20>           ;BR   2A W/C
 7791  032472                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
7792  032472  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7793  032476  061224                          61224                   ;MOVE BR TO PORT4
7794  032500  111237  002636                  MOVB    (R2),$GDDAT             ;PUT "WXPECTED" IN $GDDAT
7795  032504  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
7796  032510  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
7797  032514  001411                          BEQ     4$              ;BR IF YES
7798  032516                                  ERROR   23,YES          ;ALU ERROR
7799  032530  104455                          TRAP    C$ERDF
7800  032532  000027                          .WORD   23
7801  032534  005226                          .WORD   EM23
7802  032536  007404                          .WORD   ERR23
7803  032540                          4$:     ESCAPE  SEG
7804  032540  104410                          TRAP    C$ESCAPE
7805  032542  000014                          .WORD   10000$-.
7806  032544  005202                          INC     R2              ;NEXT DATA
7807  032546  005205                          INC     R5              ;NEXT ADDRESS
7808  032550  022705  000010                  CMP     #10,R5          ;DONE YET?
7809  032554  001324                          BNE     1$              ;BR IF NO
7810  032556                                  ENDSEG
7811  032556                          10000$:
7812  032556  104405                          TRAP    C$ESEG
7813  032560                                  EXIT    TST
7814  032560  104432                          TRAP    C$EXIT
7815  032562  000012                          .WORD   L10156-.
7816  032564     001     001     377  5$:     .BYTE   1,1,-1,-1,253,253,125,125
7817  032567     377     253     253
7818  032572     125     125
7819
7820                                  .EVEN
7821  032574                          ENDTST
7822  032574                          L10156:
7823  032574  104401                          TRAP    C$ETST
7824
7825
7826  032576                                  BADHEAD
7827                                          ;******************** TEST 69 ********************
7828                                          ;*ALU TEST
7829                                          ;*TEST OF ALU FUNCTION SUB WITH C BIT SET
7830                                          ;*ALU FUNCTION (A-B)    CODE=16
7831                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7832                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7833  032576                                  BADHEAD
7834                                          ;******************** TEST 69 ********************
7835
7836  032576                          BGNTST
7837  032576                          T69::
7838  032576                                  MYINT
7839  032576  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7840  032602                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
7841  032602  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
7842  032606  005005                          CLR     R5              ;MEM + SP ADDRESS
7843  032610  012702  032770                  MOV #5$,R2              ;POINTER TO CORRECT DATA
7844  032614  004737  003640                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7845  032620  002654                          MEMDAT                  ;POINTER TO DATA
7846  032622  004737  004012                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7847  032626  002664                          SPDAT                   ;POINTER TO DATA
```

```
7848  032630                                  BGNSEG
7849  032630  104404                          TRAP    C$BSEG
7850  032632  004737  004076          1$:     JSR     PC,SETC         ;SET C BIT!
7851  032636  042737  000017  032654          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7852  032644  050537  032654                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
7853  032650                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7854  032650  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7855  032654  010000                  2$:     010000                  ;LOAD MAR
7856  032656  042737  000017  032674          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7857  032664  050537  032674                  BIS     R5,3$           ;ADD ADDRESS  TO INSTRUCTION
7858  032670                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7859  032670  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7860  032674  040740                  3$:     040400!<16*20>          ;BR   SUB
7861  032676                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7862  032676  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7863  032702  061224                          61224                   ;MOVE BR TO PORT4
7864  032704  111237  002636                  MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
7865  032710  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
7866  032714  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
7867  032720  001411                          BEQ     4$              ;BR IF YES
7868  032722                                  ERROR   23,YES          ;ALU ERROR
7869  032734  104455                          TRAP    C$ERDF
7870  032736  000027                          .WORD   23
7871  032740  005226                          .WORD   EM23
7872  032742  007404                          .WORD   ERR23
7873  032744                          4$:     ESCAPE  SEG
7874  032744  104410                          TRAP    C$ESCAPE
7875  032746  000014                          .WORD   10000$-.
7876  032750  005202                          INC     R2                      ;NEXT DATA
7877  032752  005205                          INC     R5                      ;NEXT ADDRESS
7878  032754  022705  000010                  CMP     #10,R5                  ;DONE YET?
7879  032760  001324                          BNE     1$                      ;BR IF NO
7880  032762                                  ENDSEG
7881  032762                          10000$:
7882  032762  104405                          TRAP    C$ESEG
7883  032764                                  EXIT    TST
7884  032764  104432                          TRAP    C$EXIT
7885  032766  000012                          .WORD   L10157-.
7886  032770     000     001     377  5$:     .BYTE   0,1,-1,0,0,253,125,0
7887  032773     000     000     253
7888  032776     125     000
7889
7890                                  .EVEN
7891  033000                          ENDTST
7892  033000                          L10157:
7893  033000  104401                          TRAP    C$ETST
7894
7895
7896  033002                                  BADHEAD
7897                                  ;********************* TEST 70 *********************
7898                                  ;*ALU TEST
7899                                  ;*TEST OF ALU FUNCTION ADD W/C WITH C BIT SET
7900                                  ;*ALU FUNCTION (A PLUS B PLUS C)          CODE=01
7901                                  ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7902                                  ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7903  033002                          BADHEAD
```

```
                                                       ;******************* TEST 70 *******************

033002                                  BGNTST
033002                                  T70::
033002                                          MYINT
033002  013701  002716                          MOV     KMCSR,R1                ;GET DEVICE ADDRESS.
033006                                          MSTCLR                          ;MASTER CLEAR M8200,4,7
033006  004537  003156                          JSR     R5,.MSTCLR                      ;CLEAR M8200,4,7
033012  005005                                  CLR     R5                      ;MEM +SP ADDRESS
033014  012702  033174                          MOV     #5$,R2                  ;POINTER TO CORRECT DATA
033020  004737  003640                          JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
033024  002654                                  MEMDAT                          ;POINTER TO DATA
033026  004737  004012                          JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
033032  002664                                  SPDAT                           ;POINTER TO DATA
033034                                          BGNSEG
033034  104404                                  TRAP    C$BSEG
033036  004737  004076          1$:             JSR     PC,SETC                 ;SET C BIT!
033042  042737  000017  033060                  BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
033050  050537  033060                          BIS     R5,2$                   ;ADD ADDRESS TO INSTPUCTION
033054                                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033054  004537  003244                          JSR     R5,.ROMCLK                      ;CLOCK INSTRUCTION
033060  010000                  2$:             010000                          ;LOAD MAR
033062  042737  000017  033100                  BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
033070  050537  033100                          BIS     R5,3$                   ;ADD ADDRESS TO INSTRUCTION
033074                                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033074  004537  003244                          JSR     R5,.ROMCLK                      ;CLOCK INSTRUCTION
033100  040420                  3$:             040400!<01*20>                  ;BR - ADD W/C
033102                                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033102  004537  003244                          JSR     R5,.ROMCLK                      ;CLOCK INSTRUCTION
033106  061224                                  61224                           ;MOVE BR TO PORT4
033110  111237  002636                          MOVB    (R2),$GDDAT                     ;PUT "EXPECTED" IN $GDDAT
033114  116104  000004                          MOVB    4(R1),R4                ;PUT "FOUND" IN R4
033120  123704  002636                          CMPB    $GDDAT,R4                       ;DATA CORRECT?
033124  001411                                  BEQ     4$                      ;BR IF YES
033126                                          ERROR   23,YES                  ;ALU ERROR
033140  104455                                  TRAP    C$ERDF
033142  000027                                  .WORD   23
033144  005226                                  .WORD   EM23
033146  007404                                  .WORD   ERR23
033150                          4$:             ESCAPE  SEG
033150  104410                                  TRAP    C$ESCAPE
033152  000014                                  .WORD   10000$-.
033154  005202                                  INC     R2                      ;NEXT DATA
033156  005205                                  INC     R5                      ;NEXT ADDRESS
033160  022705  000010                          CMP     #10,R5                  ;DONE YET?
033164  001324                                  BNE     1$                      ;BR IF NO
033166                                          ENDSEG
033166                          10000$:
033166  104405                                  TRAP    C$ESEG
033170                                          EXIT    TST
033170  104432                                  TRAP    C$EXIT
033172  000012                                  .WORD   L10160-.
033174     001     000     000  5$:             .BYTE   1,0,0,-1,253,0,0,125
033177     377     253     000
033202     000     125

```

```
 7960                                   .EVEN
 7961  033204                           ENDTST
 7962  033204                           L10160:
 7963  033204  104401                           TRAP    C$ETST
 7964
 7965
 7966  033206                                   BADHEAD
 7967                                           ;******************** TEST 71 ********************
 7968                                           ;*ALU TEST
 7969                                           ;*TEST OF ALU FUNCTION SUB W/C WITH C BIT SET
 7970                                           ;*ALU FUNCTION (A-B-C)  CODE=2
 7971                                           ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 7972                                           ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 7973  033206                                   BADHEAD
 7974                                           ;******************** TEST 71 ********************
 7975
 7976
 7977  033206                           BGNTST
 7978  033206                           T71::
 7979  033206                                   MYINT
 7980  033206  013701  002716                   MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7981  033212                                   MSTCLR                  ;MASTER CLEAR M8200,4,7
 7982  033212  004537  003156                   JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 7983  033216  005005                           CLR     R5              ;MEM + SP ADDRESS
 7984  033220  012702  033400                   MOV     #5$,R2          ;POINTER TO CORRECT DATA
 7985  033224  004737  003640                   JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 7986  033230  002654                           MEMDAT                  ;POINTER TO DATA
 7987  033232  004737  004012                   JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 7988  033236  002664                           SPDAT                   ;POINTER TO DATA
 7989  033240                                   BGNSEG
 7990  033240  104404                           TRAP    C$BSEG
 7991  033242  004737  004076           1$:     JSR     PC,SETC         ;SET C BIT!
 7992  033246  042737  000017  033264           BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7993  033254  050537  033264                   BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 7994  033260                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7995  033260  004537  003244                   JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7996  033264  010000                   2$:     010000                  ;LOAD MAR
 7997  033266  042737  000017  033304           BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 7998  033274  050537  033304                   BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 7999  033300                                   ROMCLK                  ;NEXT WORD IS INSTRUCITON, ROMCLK PC=55304
 8000  033300  004537  003244                   JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8001  033304  040440                   3$:     040400!<2*20>           ;BR   SUB W/C
 8002  033306                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 8003  033306  004537  003244                   JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 8004  033312  061224                           61224                   ;MOVE BR TO PORT4
 8005  033314  111237  002636                   MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
 8006  033320  116104  000004                   MOVB    4(R1),R4        ;PUT "FOUND" IN R4
 8007  033324  123704  002636                   CMPB    $GDDAT,R4               ;DATA CORRECT?
 8008  033330  001411                           BEQ     4$              ;BR IF YES
 8009  033332                                   ERROR   23,YES          ;ALU ERROR
 8010  033344  104455                           TRAP    C$ERDF
 8011  033346  000027                           .WORD   23
 8012  033350  005226                           .WORD   EM23
 8013  033352  007404                           .WORD   ERR23
 8014  033354                           4$:     ESCAPE  SEG
 8015  033354  104410                           TRAP    C$ESCAPE
```

```
8016  033356  000014                          .WORD   10000$-.
8017  033360  005202                          INC     R2              ;NEXT DATA
8018  033362  005205                          INC     R5              ;NEXT ADDRESS
8019  033364  022705  000010                  CMP     #10,R5          ;DONE YET?
8020  033370  001324                          BNE     1$              ;BR IF NO
8021  033372                                  ENDSEG
8022  033372                          10000$:
8023  033372  104405                          TRAP    C$ESEG
8024  033374                                  EXIT    TST
8025  033374  104432                          TRAP    C$EXIT
8026  033376  000012                          .WORD   L10161-.
8027  033400     000     001     377  5$:     .BYTE   0,1,-1,0,0,253,125,0
8028  033403     000     000     253
8029  033406     125     000
8030
8031                                  .EVEN
8032  033410                          ENDTST
8033  033410                          L10161:
8034  033410  104401                          TRAP    C$ETST
8035
8036
8037  033412                                  BADHEAD
8038                                          ;******************** TEST 72 *********************
8039                                          ;*ALU TEST
8040                                          ;*TEST OF ALU FUNCTION INC A WITH C  BIT SET
8041                                          ;*ALU FUNCTION (A PLUS 1)       CODE=3
8042                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8043                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
8044  033412                                  BADHEAD
8045                                          ;******************** TEST 72 *********************
8046
8047  033412                          BGNTST
8048  033412                          T72::
8049  033412                                  MYINT
8050  033412  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
8051  033416                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
8052  033416  004537  003156                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
8053  033422  005005                          CLR     R5              ;MEM + SP ADDRESS
8054  033424  012702  033604                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
8055  033430  004737  003640                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
8056  033434  002654                          MEMDAT                  ;POINTER TO DATA
8057  033436  004737  004012                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
8058  033442  002664                          SPDAT                   ;POINTER TO DATA
8059  033444                                  BGNSEG
8060  033444  104404                          TRAP    C$BSEG
8061  033446  004737  004076          1$:     JSR     PC,SETC         ;SET C BIT!
8062  033452  042737  000017  033470          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
8063  033460  050537  033470                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
8064  033464                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
8065  033464  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8066  033470  010000                  2$:     010000                  ;LOAD MAR
8067  033472  042737  000017  033510          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
8068  033500  050537  033510                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
8069  033504                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
8070  033504  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8071  033510  040460                  3$:     040400!<3*20>           ;BR _ INC A
```

```
8072  033512                                    ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
8073  033512  004537  003244                    JSR     R5,..ROMCLK                     ;CLOCK INSTRUCTION
8074  033516  061224                            61224                           ;MOVE BR TO PORT4
8075  033520  111237  002636                    MOVB    (R2),$GDDAT                     ;PUT "EXPECTED" IN $GDDAT
8076  033524  116104  000004                    MOVB    4(R1),R4                ;PUT "FOUND IN R4
8077  033530  123704  002636                    CMPB    $GDDAT,R4                       ;DATA CORRECT?
8078  033534  001411                            BEQ     4$                      ;BR IF YES
8079  033536                                    ERROR   23,YES                  ;ALU ERROR
8080  033550  104455                            TRAP    C$ERDF
8081  033552  000027                            .WORD   23
8082  033554  005226                            .WORD   EM23
8083  033556  007404                            .WORD   ERR23
8084  033560                            4$:     ESCAPE  SEG
8085  033560  104410                            TRAP    C$ESCAPE
8086  033562  000014                            .WORD   10000$-.
8087  033564  005202                            INC     R2                      ;NEXT DATA
8088  033566  005205                            INC     R5                      ;NEXT ADDRESS
8089  033570  022705  000010                    CMP     #10,R5                  ;DONE YET?
8090  033574  001324                            BNE     1$                      ;BR IF NO
8091  033576                                    ENDSEG
8092  033576                            10000$:
8093  033576  104405                            TRAP    C$ESEG
8094  033600                                    EXIT    TST
8095  033600  104432                            TRAP    C$EXIT
8096  033602  000012                            .WORD   L10162-.
8097  033604     001     001     000  5$:       .BYTE   1,1,0,0,126,126,253,253
8098  033607     000     126     126
8099  033612     253     253
8100
8101                                    .EVEN
8102  033614                            ENDTST
8103  033614                            L10162:
8104  033614  104401                            TRAP    C$ETST
8105
8106
8107  033616                                    BADHEAD
8108                                            ;******************** TEST 73 ********************
8109                                            ;*ALU TEST
8110                                            ;*TEST OF ALU FUNCTION 2A WITH C BIT SET
8111                                            ;*ALU FUNCTION (A PLUS A)        CODE=5
8112                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8113                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
8114  033616                                    BADHEAD
8115                                            ;******************** TEST 73 ********************
8116
8117  033616                            BGNTST
8118  033616                            T73::
8119  033616                                    MYINT
8120  033616  013701  002716                    MOV     KMCSR,R1                ;GET DEVICE ADDRESS.
8121  033622                                    MSTCLR                          ;MASTER CLEAR M8200,4,7
8122  033622  004537  003156                    JSR     R5,..MSTCLR                     ;CLEAR M8200,4,7
8123  033626  005005                            CLR     R5                      ;MEM + SP ADDRESS
8124  033630  012702  034010                    MOV     #5$,R2                  ;POINTER TO CORRECT DATA
8125  033634  004737  003640                    JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
8126  033640  002654                            MEMDAT                          ;POINTER TO DATA
8127  033642  004737  004012                    JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
```

```
                                                SPDAT                   ;POINTER TO DATA
                                                BGNSEG
                                                TRAP    C$BSEG
                                        1$:     JSR     PC,SETC         ;SET 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$:     040400!<5*20>           ;BR    2A
                                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
                                                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
                                                61224                   ;MOVE BR TO PORT4
                                                MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
                                                MOVB    4(R1),R4        ;PUT "FOUND IN R4
                                                CMPB    $GDDAT,R4               ;DATA CORRECT?
                                                BEQ     4$              ;BR IF YES
                                                ERROR   23,YES          ;ALU ERROR
                                                TRAP    C$ERDF
                                                .WORD   23
                                                .WORD   EM23
                                                .WORD   ERR23
                                        4$:     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$EXIT
                                                .WORD   L10163-.
                000     000     376     5$:     .BYTE   0,0,376,376,252,252,124,124
                376     252     252
                124     124

                                        .EVEN
                                        ENDTST
                                        L10163:
                                                TRAP    C$ETST


                                                BADHEAD
                                                ;******************** TEST 74 ********************
                                                ;*ALU TEST
                                                ;*TEST OF ALU FUNCTION A PLUS C WITH C BIT SET
                                                ;*ALU FUNCTION (A PLUS C)        CODE=4
                                                ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
```

```
8184  034022                                        BADHEAD
8185                                                ;******************** TEST 74 ********************
8186
8187  034022                                  BGNTST
8188  034022                                  T74::
8189  034022                                        MYINT
8190  034022  013701  002716                        MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
8191  034026                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
8192  034026  004537  003156                        JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
8193  034032  005005                                CLR     R5              ;MEM + SP ADDRESS
8194  034034  012702  034214                        MOV     #5$,R2          ;POINTER TO CORRECT DATA
8195  034040  004737  003640                        JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
8196  034044  002654                                MEMDAT                  ;POINTER TO DATA
8197  034046  004737  004012                        JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
8198  034052  002664                                SPDAT                   ;POINTER TO DATA
8199  034054                                        BGNSEG
8200  034054  104404                                TRAP    C$BSEG
8201  034056  004737  004076                  1$:   JSR     PC,SETC         ;SET C BIT!
8202  034062  042737  000017  034100                BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
8203  034070  050537  034100                        BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
8204  034074                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
8205  034074  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8206  034100  010000                          2$:   010000                  ;LOAD MAR
8207  034102  042737  000017  034120                BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
8208  034110  050537  034120                        BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
8209  034114                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
8210  034114  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8211  034120  040500                          3$:   040400!<4*20>           ;BR   A PLUS C
8212  034122                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
8213  034122  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8214  034126  061224                                61224                   ;MOVE BR TO PORT4
8215  034130  111237  002636                        MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
8216  034134  116104  000004                        MOVB    4(R1),R4        ;PUT "FOUND IN R4
8217  034140  123704  002636                        CMPB    $GDDAT,R4               ;DATA CORRECT?
8218  034144  001411                                BEQ     4$              ;BR IF YES
8219  034146                                        ERROR   23,YES          ;ALU ERROR
8220  034160  104455                                TRAP    C$ERDF
8221  034162  000027                                .WORD   23
8222  034164  005226                                .WORD   EM23
8223  034166  007404                                .WORD   ERR23
8224  034170                                  4$:   ESCAPE  SEG
8225  034170  104410                                TRAP    C$ESCAPE
8226  034172  000014                                .WORD   10000$-.
8227  034174  005202                                INC     R2              ;NEXT DATA
8228  034176  005205                                INC     R5              ;NEXT ADDRESS
8229  034200  022705  000010                        CMP     #10,R5          ;DONE YET?
8230  034204  001324                                BNE     1$              ;BR IF NO
8231  034206                                        ENDSEG
8232  034206                                  10000$:
8233  034206  104405                                TRAP    C$ESEG
8234  034210                                        EXIT    TST
8235  034210  104432                                TRAP    C$EXIT
8236  034212  000012                                .WORD   L10164-.
8237  034214     001     001     000           5$:  .BYTE   1,1,0,0,126,126,253,253
8238  034217     000     126     126
8239  034222     253     253
```

```
8240
8241                                    .EVEN
8242  034224                            ENDTST
8243  034224                            L10164:
8244  034224  104401                            TRAP    C$ETST
8245
8246  034226                                    BADHEAD
8247                                              ;********** TEST 75 **********
8248                                              ;*ALU TEST
8249                                              ;*TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT SET
8250                                              ;*ALU FUNCTION (A-B-1)  CODE=17
8251                                              ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8252                                              ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
8253  034226                                    BADHEAD
8254                                              ;********** TEST 75 **********
8255
8256  034226                            BGNTST
8257  034226                            T75::
```

```
8258  034226                                  MYINT
8259  034226  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
8260  034232                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
8261  034232  004537  003156                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
8262  034236  005005                          CLR     R5              ;MEM + SP ADDRESS
8263  034240  012702  034420                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
8264  034244  004737  003640                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
8265  034250  002654                          MEMDAT                  ;POINTER TO DATA
8266  034252  004737  004012                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
8267  034256  002664                          SPDAT                   ;POINTER TO DATA
8268  034260                                  BGNSEG
8269  034260  104404                          TRAP    C$BSEG
8270  034262  004737  004076          1$:     JSR     PC,SETC         ;SET C BIT!
8271  034266  042737  000017  034304          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
8272  034274  050537  034304                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
8273  034300                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
8274  034300  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8275  034304  010000                  2$:     010000                  ;LOAD MAR
8276  034306  042737  000017  034324          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
8277  034314  050537  034324                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
8278  034320                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
8279  034320  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8280  034324  040760                  3$:     040400!<17*20>          ;BR   2'S COMP SUB
8281  034326                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
8282  034326  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
8283  034332  061224                          61224                   ;MOVE BR TO PORT4
8284  034334  111237  002636                  MOVB    (R2),$GDDAT             ;PUT 'EXPECTED" IN $GDDAT
8285  034340  116104  000004                  MOVB    4(R1),R4        ;PUT 'FOUND IN R4
8286  034344  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
8287  034350  001411                          BEQ     4$              ;BR IF YES
8288  034352                                  ERROR   23,YES          ;ALU ERROR
8289  034364  104455                          TRAP    C$ERDF
8290  034366  000027                          .WORD   23
8291  034370  005226                          .WORD   EM23
8292  034372  007404                          .WORD   ERR23
8293  034374                          4$:     ESCAPE  SEG
8294  034374  104410                          TRAP    C$ESCAPE
8295  034376  000014                          .WORD   10000$-.
8296  034400  005202                          INC     R2              ;NEXT DATA
8297  034402  005205                          INC     R5              ;NEXT ADDRESS
8298  034404  022705  000010                  CMP     #10,R5          ;DONE YET?
8299  034410  001324                          BNE     1$              ;BR IF NO
8300  034412                                  ENDSEG
8301  034412                          10000$:
8302  034412  104405                          TRAP    C$ESEG
8303  034414                                  EXIT    TST
8304  034414  104432                          TRAP    C$EXIT
8305  034416  000012                          .WORD   L10165-.
8306  034420     377     000     376  5$:     .BYTE   -1,0,376,-1,-1,252,124,-1
8307  034423     377     377     252
8308  034426     124     377
8309
8310                                  .EVEN
8311  034430                          ENDTST
8312  034430                          L10165:
8313  034430  104401                          TRAP    C$ETST
```

```
8314
8315
8316  034432                                  BADHEAD
8317                                          ;******************** TEST 76 ********************
8318                                          ;*ALU TEST
8319                                          ;*TEST OF ALU FUNCTION DEC A WITH C BIT SET
8320                                          ;*ALU FUNCTION (A-1)    CODE=7
8321                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8322                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
8323  034432                                  BADHEAD
8324                                          ;******************** TEST 76 ********************
8325
8326  034432                          BGNTST
8327  034432                          T76::
8328  034432                                  MYINT
8329  034432  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
8330  034436                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
8331  034436  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
8332  034442  005005                          CLR     R5              ;MEM + SP ADDRESS
8333  034444  012702  034624                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
8334  034450  004737  003640                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
8335  034454  002654                          MEMDAT                  ;POINTER TO DATA
8336  034456  004737  004012                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
8337  034462  002664                          SPDAT                   ;POINTER TO DATA
8338  034464                                  BGNSEG
8339  034464  104404                          TRAP    C$BSEG
8340  034466  004737  004076          1$:     JSR     PC,SETC         ;SET C BIT!
8341  034472  042737  000017  034510          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
8342  034500  050537  034510                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
8343  034504                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
8344  034504  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
8345  034510  010000                  2$:     010000                  ;LOAD MAR
8346  034512  042737  000017  034530          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
8347  034520  050537  034530                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
8348  034524                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
8349  034524  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
8350  034530  040560                  3$:     040400!<7*20>           ;BR   DEC A
8351  034532                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
8352  034532  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
8353  034536  061224                          61224                   ;MOVE BR TO PORT4
8354  034540  111237  002636                  MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
8355  034544  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND IN R4
8356  034550  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
8357  034554  001411                          BEQ     4$              ;BR IF YES
8358  034556                                  ERROR   23,YES          ;ALU ERROR
8359  034570  104455                          TRAP    C$ERDF
8360  034572  000027                          .WORD   23
8361  034574  005226                          .WORD   EM23
8362  034576  007404                          .WORD   ERR23
8363  034600                          4$:     ESCAPE  SEG
8364  034600  104410                          TRAP    C$ESCAPE
8365  034602  000014                          .WORD   10000$-.
8366  034604  005202                          INC     R2              ;NEXT DATA
8367  034606  005205                          INC     R5              ;NEXT ADDRESS
8368  034610  022705  000010                  CMP     #10,R5          ;DONE YET?
8369  034614  001324                          BNE     1$              ;BR IF NO
```

```
 8370  034616                                    ENDSEG
 8371  034616                              10000$:
 8372  034616  104405                              TRAP    C$ESEG
 8373  034620                                      EXIT    TST
 8374  034620  104432                              TRAP    C$EXIT
 8375  034622  000012                              .WORD   L10166-.
 8376  034624     377     377     376  5$:       .BYTE   -1,-1,376,376,124,124,251,251
 8377  034627     376     124     124
 8378  034632     251     251
 8379
 8380                                        .EVEN
 8381  034634                              ENDTST
 8382  034634                              L10166:
 8383  034634  104401                              TRAP    C$ETST
 8384
 8385
```

```
8386
8387
8388
8389
8390
8391
```

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

034636                                  BGNHRD
034636  000022                          .WORD L10167-L$HARD/2
034640                          L$HARD::

034640                                  GPRMD   WMP,0,0,7,0,7,YES
034640  000032                          .WORD   T$CODE
034642  034704                          .WORD   WMP
034644  000007                          .WORD   7
034646  000000                          .WORD   T$LOLIM
034650  000007                          .WORD   T$HILIM
034652                                  GPRMA   ADDRES,2,0,160000,177776,YES
034652  001031                          .WORD   T$CODE
034654  034763                          .WORD   ADDRES
034656  160000                          .WORD   T$LOLIM
034660  177776                          .WORD   T$HILIM
034662                                  GPRMA   VECTOR,4,0,0,770,YES
034662  002031                          .WORD   T$CODE
034664  035022                          .WORD   VECTOR
034666  000000                          .WORD   T$LOLIM
034670  000770                          .WORD   T$HILIM
034672                                  GPRMD   PRIRTY,6,0,7000,4,7,YES
034672  003032                          .WORD   T$CODE
034674  035064                          .WORD   PRIRTY
034676  007000                          .WORD   7000
034700  000004                          .WORD   T$LOLIM
034702  000007                          .WORD   T$HILIM
                                ;       GPRMD   LNUNIT,10,0,3,0,3,YES
                                ;       GPRMD   SWPAC1,12,0,377,0,377,YES
                                ;       GPRMD   SWPAC2,14,0,377,0,377,YES
                                ;       GPRMD   LOOPBK,16,0,40000,0,1,YES

034704                                  ENDHRD
                                        .EVEN
034704                          L10167:

034704  044127  041511  020110  WMP:    .ASCIZ  'WHICH MICRO-PROCESSOR:'
034712  044515  051103  026517
034720  051120  041517  051505
034726  047523  035122     000
034733     060  046475  031070          .ASCIZ  '0=M8200,4=M8204,7=M8207'
034740  030060  032054  046475
034746  031070  032060  033454
034754  046475  031070  033460
```

```
 8448  034762     000
 8449  034763     115  041511  047522  ADDRES: .ASCIZ  /MICRO-PROCESSOR CSR ADDRESS : /
 8450  034770  050055  047522  042503
 8451  034776  051523  051117  041440
 8452  035004  051123  040440  042104
 8453  035012  042522  051523  035040
 8454  035020  000040
 8455  035022  044515  051103  026517  VECTOR: .ASCIZ  /MICRO-PROCESSOR VECTOR ADDRESS : /
 8456  035030  051120  041517  051505
 8457  035036  047523  020122  042526
 8458  035044  052103  051117  040440
 8459  035052  042104  042522  051523
 8460  035060  035040  000040
 8461  035064  044515  051103  026517  PRIRTY: .ASCIZ  /MICRO-PROCESSOR PRIORITY LEVEL : /
 8462  035072  051120  041517  051505
 8463  035100  047523  020122  051120
 8464  035106  047511  044522  054524
 8465  035114  046040  053105  046105
 8466  035122  035040  000040
 8467  035126  044127  041511  020110  LNUNIT: .ASCIZ  /WHICH LINE UNIT (0-3)? 0=NONE,1=M8201,2=M8202,3=M8203 : /
 8468  035134  044514  042516  052440
 8469  035142  044516  020124  030050
 8470  035150  031455  037451  030040
 8471  035156  047075  047117  026105
 8472  035164  036461  034115  030062
 8473  035172  026061  036462  034115
 8474  035200  030062  026062  036463
 8475  035206  034115  030062  020063
 8476  035214  020072     000
 8477  035217     123  044527  041524  SWPAC1: .ASCIZ  /SWITCH PACK #1 (DDCMP LINE #) : /
 8478  035224  020110  040520  045503
 8479  035232  021440  020061  042050
 8480  035240  041504  050115  046040
 8481  035246  047111  020105  024443
 8482  035254  035040  000040
 8483  035260  053523  052111  044103  SWPAC2: .ASCIZ  /SWITCH PACK #2 (BM873 BOOT ADR) : /
 8484  035266  050040  041501  020113
 8485  035274  031043  024040  046502
 8486  035302  033470  020063  047502
 8487  035310  052117  040440  051104
 8488  035316  020051  020072     000
 8489  035323     127  046111  020114  LOOPBK: .ASCIZ  /WILL TEST CONNECTOR(S) BE USED ? 0=NO,1=YES : /
 8490  035330  042524  052123  041440
 8491  035336  047117  042516  052103
 8492  035344  051117  051450  020051
 8493  035352  042502  052440  042523
 8494  035360  020104  020077  036460
 8495  035366  047516  030454  054475
 8496  035374  051505  035040  000040
 8497
 8498                                  .EVEN
 8499
 8500
 8501
 8502
 8503
```

```
8504
```

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

035402                          BGNSFT
035402  000000                  .WORD L10170-L$SOFT/2
035404                  L$SOFT::

035404                          ENDSFT
                                .EVEN
035404                  L10170:

                        .EVEN




035404                          ENDMOD


035404                  CORMAX:

035404  000000                  .WORD   0               ;START OF NPR AREA (TEST 55)
        037776          .=37776
037776  000000          MEMEND: .WORD   0               ;END OF NPR AREA
040000                          LASTAD
                                .EVEN
040000  000000                  .WORD   0
040002  000000                  .WORD   0
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  034763            8417    8449#
ADR   = 000020 G          1719#
ASSEMB= 000010            1368
BIT0  = 000001 G          1692#   2008    2009    3420    6041
BIT00 = 000001 G          1681#   1692
BIT01 = 000002 G          1680#   1691
BIT02 = 000004 G          1679#   1690
BIT03 = 000010 G          1678#   1689
BIT04 = 000020 G          1677#   1688
BIT05 = 000040 G          1676#   1687
BIT06 = 000100 G          1675#   1686
BIT07 = 000200 G          1674#   1685
BIT08 = 000400 G          1673#   1684
BIT09 = 001000 G          1672#   1683
BIT1  = 000002 G          1691#   2006    2008    2009
BIT10 = 002000 G          1671#   2091    2099    3884    5818    5885
BIT11 = 004000 G          1670#   3532    3550
BIT12 = 010000 G          1669#   3130    3582    3600
BIT13 = 020000 G          1668#   2094    2102    3141
BIT14 = 040000 G          1667#
BIT15 = 100000 G          1666#   3145
BIT2  = 000004 G          1690#   2009
BIT3  = 000010 G          1689#
BIT4  = 000020 G          1688#
BIT5  = 000040 G          1687#
BIT6  = 000100 G          1686#   1991    1992
BIT7  = 000200 G          1685#   1992
BIT8  = 000400 G          1684#   2072    3944    5820    5887
BIT9  = 001000 G          1683#   3479    3497    3884    3942    3944    5818    5885
BOE   = 000400 G          1723#
CLKX    002570            1792#
CLRALL  003274            2012#
CLRC    004060            2214#   6174    6243    6312    6381    6450    6519    6588    6657    6726    6797    6867    6938
                          7008    7078    7148    7219
CORMAX  035404            6037    6076    8535#
CRAMA   003522            2100    2109#
CZDMP   002000 G          1374#
C$AU  = 000052            1368#   3255
C$AUTO= 000061            1368#   3201
C$BRK = 000022            1368#
C$BSEG= 000004            1368#   3359    3383    3422    3477    3496    3530    3549    3580    3599    3631    3690    3712
                          3754    3776    3819    3843    3883    3903    3982    4011    4060    4089    4138    4167    4216
                          4245    4294    4328    4382    4416    4470    4499    4548    4577    4626    4655    4704    4733
                          4782    4811    4860    4889    4938    4967    5016    5045    5094    5132    5179    5208    5263
                          5298    5362    5400    6112    6173    6242    6311    6380    6449    6518    6587    6656    6725
                          6796    6866    6937    7007    7077    7147    7218    7288    7359    7429    7499    7569    7639
                          7709    7779    7849    7919    7990    8060    8130    8200    8269    8339
C$BSUB= 000002            1368#   5261
C$CEFG= 000045            1368#
C$CLCK= 000062            1368#
C$CLEA= 000012            1368#   3217
C$CLOS= 000035            1368#
C$CLP1= 000006            1368#
C$CVEC= 000036            1368#
C$DCLN= 000044            1368#
C$DODU= 000051            1368#   3195
```

```
C$DRPT= 000024   1368#
C$DU  = 000053   1368#  3237
C$EDIT= 000003   1368#  1446
C$ERDF= 000055   1368#  2184  3292  3321  3366  3389  3430  3442  3485  3502  3538  3555  3588
                 3605   3640  3652  3697  3720  3761  3785  3826  3849  3891  3909  3951  3995
                 4025   4073  4103  4151  4181  4229  4259  4311  4346  4399  4434  4483  4513
                 4561   4591  4639  4669  4717  4747  4795  4825  4873  4903  4951  4981  5029
                 5059   5115  5143  5192  5223  5283  5320  5383  5415  5463  5469  5508  5514
                 5568   5618  5626  5674  5726  5776  5847  5901  5950  5965  6012  6060  6131
                 6193   6262  6331  6400  6469  6538  6607  6676  6745  6816  6886  6957  7027
                 7097   7167  7238  7308  7379  7449  7519  7589  7659  7729  7799  7869  7939
                 8010   8080  8150  8220  8289  8359
C$ERHR= 000056   1368#
C$ERRO= 000060   1368#
C$ERSF= 000054   1368#
C$ERSO= 000057   1368#
C$ESCA= 000010   1368#  3283  3300  3371  3394  3435  3447  3490  3507  3543  3593  3645  3657
                 3702   3725  3766  3790  3831  3854  3896  3914  3957  4000  4030  4078  4108
                 4156   4186  4234  4264  4316  4351  4404  4439  4488  4518  4566  4596  4644
                 4674   4722  4752  4800  4830  4878  4908  4956  4986  5034  5064  5120  5148
                 5197   5228  5288  5325  5388  5420  5679  5731  5781  6065  6136  6198  6267
                 6336   6405  6474  6543  6612  6681  6750  6821  6891  6962  7032  7102  7172
                 7243   7313  7384  7454  7524  7594  7664  7734  7804  7874  7944  8015  8085
                 8155   8225  8294  8364
C$ESEG= 000005   1368#  3379  3402  3453  3494  3511  3547  3561  3597  3611  3663  3709  3733
                 3773   3798  3841  3862  3900  3918  4007  4038  4085  4116  4163  4194  4241
                 4272   4324  4360  4412  4448  4495  4526  4573  4604  4651  4682  4729  4760
                 4807   4838  4885  4916  4963  4994  5041  5072  5128  5157  5204  5238  5295
                 5333   5396  5428  6140  6206  6275  6344  6413  6482  6551  6620  6689  6758
                 6829   6899  6970  7040  7110  7180  7251  7321  7392  7462  7532  7602  7672
                 7742   7812  7882  7952  8023  8093  8163  8233  8302  8372
C$ESUB= 000003   1368#  5340
C$ETST= 000001   1368#  3304  3328  3405  3456  3515  3565  3615  3666  3736  3801  3865  3921
                 3962   4041  4119  4197  4275  4363  4451  4529  4607  4685  4763  4841  4919
                 4997   5075  5160  5241  5343  5431  5477  5522  5577  5636  5688  5740  5790
                 5854   5911  5973  6019  6088  6143  6217  6286  6355  6424  6493  6562  6631
                 6700   6770  6840  6911  6981  7051  7121  7191  7262  7332  7403  7473  7543
                 7613   7683  7753  7823  7893  7963  8034  8104  8174  8244  8313  8383
C$EXIT= 000032   1368#  5682  5734  5784  6021  6208  6277  6346  6415  6484  6553  6622  6691
                 6760   6831  6901  6972  7042  7112  7182  7253  7323  7394  7464  7534  7604
                 7674   7744  7814  7884  7954  8025  8095  8165  8235  8304  8374
C$GETB= 000026   1368#
C$GETW= 000027   1368#
C$GMAN= 000043   1368#
C$GPHR= 000042   1368#  3093
C$GPLO= 000030   1368#
C$GPRI= 000040   1368#
C$INIT= 000011   1368#  3175
C$INLP= 000020   1368#
C$MANI= 000050   1368#
C$MEM = 000031   1368#
C$MSG = 000023   1368#  2507  2524  2541  2559  2577  2595  2613  2631  2648  2665  2675  2692
                 2709   2719  2729  2746  2756  2766  2783  2793  2810  2827  2845  2862  2880
                 2897   2907  2917  2934  2951  2968  2985  3005
C$OPEN= 000034   1368#
C$PNTB= 000014   1368#  2497  2504  2514  2521  2531  2538  2548  2556  2566  2574  2584  2592
```

```
                        2602    2610    2620    2628    2638    2645    2655    2662    2672    2682    2689    2699    2706
                        2716    2726    2736    2743    2753    2763    2773    2780    2790    2800    2807    2817    2824
                        2834    2842    2852    2859    2869    2877    2887    2894    2904    2914    2924    2931    2941
                        2948    2958    2965    2975    2982
C$PNTF= 000017           1368#   2994    3001
C$PNTS= 000016           1368#
C$PNTX= 000015           1368#
C$QIO = 000377           1368#
C$RDBU= 000007           1368#
C$REFG= 000047           1368#   3058    3064    3070    3077
C$RESE= 000033           1368#   3213    3234    5445    5566    5648
C$REVI= 000003           1368#   1445
C$RFLA= 000021           1368#
C$RPT = 000025           1368#   3028
C$SEFG= 000046           1368#
C$SPRI= 000041           1368#   5453    5460    5498    5505    5541    5559    5596    5615
C$SVEC= 000037           1368#
C$TPRI= 000013           1368#
DFPTBL  002364 G         1600#
DH0     005716           2429#   2668    2712    2722    2749    2759    2786    2900    2910
DH1     005717           2430#   2544    2562
DH2     005750           2435#   2493    2510    2527    2634    2651    2678    2732    2796    2813    2848    2883    2920
                         2937    2954    2971
DH27    006030           2443#   2830    2865
DH3     005764           2437#   2580    2598    2616
DH4     006014           2441#   2695    2769
DIAGMC= 000000           1368
EF.CON= 000036 G         1699#   3069
EF.NEW= 000035 G         1700#   3063
EF.PWR= 000034 G         1701#
EF.RES= 000037 G         1698#   3076
EF.STA= 000040 G         1697#   3057
EM1     004261           2274#   2990
EM10    004622           2315#
EM11    004654           2320#   5676    5728    5778    5849    5903    6062
EM12    004665           2322#
EM13    004707           2326#   5952    5967    6014
EM14    004731           2330#
EM15    004754           2334#   6195    6264    6333    6402    6471    6540    6609    6678    6747    6818    6888    6959
                         7029    7099    7169    7240    7451
EM16    005017           2340#
EM17    005064           2348#
EM2     004307           2278#   3391    3444    3828    3851
EM20    005113           2352#
EM21    005160           2359#
EM22    005207           2363#
EM23    005226           2366#   7310    7381    7521    7591    7661    7731    7801    7871    7941    8012    8082    8152
                         8222    8291    8361
EM24    005267           2372#
EM25    005453           2394#
EM26    005474           2397#   3323    3368    3487    3504    3540    3557    3590    3607    3893    3911
EM27    005525           2402#   3432    3642    3654    3699    3722    3763    3787    3997    4027    4075    4105    4153
                         4183    4231    4261    4313    4348    4401    4436
EM28    005554           2406#   3953
EM29    005605           2411#   4485    4515    4563    4593    4641    4671    4719    4749    4797    4827    4875    4905
                         4953    4983    5031    5061    5117
```

```
EM3     004416            2291#  5194   5225
EM30    004353            2285#  5145
EM31    005312            2376#  5465   5510   5620
EM32    005341            2380#  5471   5516   5628
EM33    005400            2386#  5570
EM34    005435            2391#  6133
EM35    005064            2347#
EM36    005640            2186   2417#
EM37    005715            2425#  3294
EM4     004444            2295#  5285   5322   5385
EM5     004472            2299#  5417
EM6     004533            2305#
EM7     004561            2309#
ENDBUG  003244            2000#
ENDIT   011254            3072   3163   3167   3170   3172#
ERRFLG  002560            1788#
ERR1    006054 G          2492#
ERR10   006576 G          2615#
ERR11   006660 G          2633#  5677   5729   5779   5850   5904   6063
ERR12   006736 G          2650#
ERR13   007014 G          2667#  5953   5968   6015
ERR14   007042 G          2677#
ERR15   007120 G          2694#  6196   6265   6334   6403   6472   6541   6610   6679   6748   6819   6889   6960
                          7030   7100   7170   7241   7452
ERR16   007176 G          2711#
ERR17   007224 G          2721#
ERR2    006132 G          2509#  3392   3445   3829   3852
ERR20   007252 G          2731#
ERR21   007330 G          2748#
ERR22   007356 G          2758#
ERR23   007404 G          2768#  7311   7382   7522   7592   7662   7732   7802   7872   7942   8013   8083   8153
                          8223   8292   8362
ERR24   007462 G          2785#
ERR25   007510 G          2795#
ERR26   007566 G          2812#  3324   3369   3488   3505   3541   3558   3591   3608   3894   3912
ERR27   007644 G          2829#  3433   3643   3655   3700   3723   3764   3788   3998   4028   4076   4106   4154
                          4184   4232   4262   4314   4349   4402   4437
ERR28   007726 G          2847#  3954
ERR29   010004 G          2864#  4486   4516   4564   4594   4642   4672   4720   4750   4798   4828   4876   4906
                          4954   4984   5032   5062   5118
ERR3    006210 G          2526#  5195   5226
ERR30   010066 G          2882#  5146
ERR31   010144 G          2899#  5466   5511   5621
ERR32   010172 G          2909#  5472   5517   5629
ERR33   010220 G          2919#  5571
ERR34   010276 G          2936#  6134
ERR35   010354 G          2953#
ERR36   010432 G          2187   2970#
ERR37   010510 G          2988#  3295
ERR4    006266 G          2543#  5286   5323   5386
ERR5    006350 G          2561#  5418
ERR6    006432 G          2579#
ERR7    006514 G          2597#
EVL     =  000004 G       1717#
ESEND   =  002100         1368#
ESLOAD  =  000035         1368#  1469
```

```
FLAG    002620          1E05#
FM1     004114          2249#  2494  2511  2528  2545  2563  2581  2599  2617  2635  2652  2669  2679
                        2696   2713  2723  2733  2750  2760  2770  2787  2797  2814  2831  2849  2866
                        2884   2901  2911  2921  2938  2955  2972  2991
FTIME   002646          1816#  3048  3052*
F$AU  = 000015          1368#  3252  3254
F$AUTO= 000020          1368#  3179  3200
F$BGN = 000040          1368#  1374  1490  2492  2509  2526  2543  2561  2579  2597  2615  2633  2650
                        2667   2677  2694  2711  2721  2731  2748  2758  2768  2785  2795  2812  2829
                        2847   2864  2882  2899  2909  2919  2936  2953  2970  2988  3018  3024  3042
                        3179   3211  3231  3252  3275  3283  3300  3303  3313  3327  3351  3359  3371
                        3383   3394  3404  3416  3422  3435  3447  3455  3473  3477  3490  3496  3507
                        3514   3526  3530  3543  3549  3564  3576  3580  3593  3599  3614  3625  3631
                        3645   3657  3665  3683  3690  3702  3712  3725  3735  3747  3754  3766  3776
                        3790   3800  3812  3819  3831  3843  3854  3864  3876  3883  3896  3903  3914
                        3920   3936  3957  3961  3973  3982  4000  4011  4030  4040  4052  4060  4078
                        4089   4108  4118  4130  4138  4156  4167  4186  4196  4208  4216  4234  4245
                        4264   4274  4286  4294  4316  4328  4351  4362  4374  4382  4404  4416  4439
                        4450   4462  4470  4488  4499  4518  4528  4540  4548  4566  4577  4596  4606
                        4618   4626  4644  4655  4674  4684  4696  4704  4722  4733  4752  4762  4774
                        4782   4800  4811  4830  4840  4852  4860  4878  4889  4908  4918  4930  4938
                        4956   4967  4986  4996  5008  5016  5034  5045  5064  5074  5086  5094  5120
                        5132   5148  5159  5171  5179  5197  5208  5228  5240  5252  5260  5263  5288
                        5298   5325  5339  5342  5354  5362  5388  5400  5420  5430  5441  5476  5487
                        5521   5533  5576  5588  5635  5646  5679  5682  5687  5698  5731  5734  5739
                        5750   5781  5784  5789  5801  5853  5869  5910  5922  5972  5984  6018  6030
                        6065   6081  6087  6102  6112  6136  6142  6161  6173  6198  6208  6216  6230
                        6242   6267  6277  6285  6299  6311  6336  6346  6354  6368  6380  6405  6415
                        6423   6437  6449  6474  6484  6492  6506  6518  6543  6553  6561  6575  6587
                        6612   6622  6630  6644  6656  6681  6691  6699  6713  6725  6750  6760  6769
                        6784   6796  6821  6831  6839  6854  6866  6891  6901  6910  6925  6937  6962
                        6972   6980  6995  7007  7032  7042  7050  7065  7077  7102  7112  7120  7135
                        7147   7172  7182  7190  7206  7218  7243  7253  7261  7276  7288  7313  7323
                        7331   7347  7359  7384  7394  7402  7417  7429  7454  7464  7472  7487  7499
                        7524   7534  7542  7557  7569  7594  7604  7612  7627  7639  7664  7674  7682
                        7697   7709  7734  7744  7752  7767  7779  7804  7814  7822  7837  7849  7874
                        7884   7892  7907  7919  7944  7954  7962  7978  7990  8015  8025  8033  8048
                        8060   8085  8095  8103  8118  8130  8155  8165  8173  8188  8200  8225  8235
                        8243   8257  8269  8294  8304  8312  8327  8339  8364  8374  8382  8406  8518
                        8533
F$CLEA= 000007          1368#  3211  3216
F$DU  = 000016          1368#  3231  3236
F$END = 000041          1368#  1374  2508  2525  2542  2560  2578  2596  2614  2632  2649  2666  2676
                        2693   2710  2720  2730  2747  2757  2767  2784  2794  2811  2828  2846  2863
                        2881   2898  2908  2918  2935  2952  2969  2986  3006  3022  3029  3176  3202
                        3218   3238  3256  3275  3283  3300  3303  3305  3313  3327  3329  3351  3371
                        3380   3394  3403  3404  3406  3416  3435  3447  3454  3455  3457  3473  3490
                        3495   3507  3512  3514  3516  3526  3543  3548  3562  3564  3566  3576  3593
                        3598   3612  3614  3616  3625  3645  3657  3664  3665  3667  3683  3702  3710
                        3725   3734  3735  3737  3747  3766  3774  3790  3799  3800  3802  3812  3831
                        3842   3854  3863  3864  3866  3876  3896  3901  3914  3919  3920  3922  3936
                        3957   3961  3963  3973  4000  4008  4030  4039  4040  4042  4052  4078  4086
                        4108   4117  4118  4120  4130  4156  4164  4186  4195  4196  4198  4208  4234
                        4242   4264  4273  4274  4276  4286  4316  4325  4351  4361  4362  4364  4374
                        4404   4413  4439  4449  4450  4452  4462  4488  4496  4518  4527  4528  4530
                        4540   4566  4574  4596  4605  4606  4608  4618  4644  4652  4674  4683  4684
```

```
                    4686  4696  4722  4730  4752  4761  4762  4764  4774  4800  4808  4830  4839
                    4840  4842  4852  4878  4886  4908  4917  4918  4920  4930  4956  4964  4986
                    4995  4996  4998  5008  5034  5042  5064  5073  5074  5076  5086  5120  5129
                    5148  5158  5159  5161  5171  5197  5205  5228  5239  5240  5242  5252  5260
                    5288  5296  5325  5334  5339  5341  5342  5344  5354  5388  5397  5420  5429
                    5430  5432  5441  5476  5478  5487  5521  5523  5533  5576  5578  5588  5635
                    5637  5646  5679  5682  5687  5689  5698  5731  5734  5739  5741  5750  5781
                    5784  5789  5791  5801  5853  5855  5869  5910  5912  5922  5972  5974  5984
                    6018  6020  6030  6065  6081  6087  6089  6102  6136  6141  6142  6144  6161
                    6198  6207  6208  6216  6218  6230  6267  6276  6277  6285  6287  6299  6336
                    6345  6346  6354  6356  6368  6405  6414  6415  6423  6425  6437  6474  6483
                    6484  6492  6494  6506  6543  6552  6553  6561  6563  6575  6612  6621  6622
                    6630  6632  6644  6681  6690  6691  6699  6701  6713  6750  6759  6760  6769
                    6771  6784  6821  6830  6831  6839  6841  6854  6891  6900  6901  6910  6912
                    6925  6962  6971  6972  6980  6982  6995  7032  7041  7042  7050  7052  7065
                    7102  7111  7112  7120  7122  7135  7172  7181  7182  7190  7192  7206  7243
                    7252  7253  7261  7263  7276  7313  7322  7323  7331  7333  7347  7384  7393
                    7394  7402  7404  7417  7454  7463  7464  7472  7474  7487  7524  7533  7534
                    7542  7544  7557  7594  7603  7604  7612  7614  7627  7664  7673  7674  7682
                    7684  7697  7734  7743  7744  7752  7754  7767  7804  7813  7814  7822  7824
                    7837  7874  7883  7884  7892  7894  7907  7944  7953  7954  7962  7964  7978
                    8015  8024  8025  8033  8035  8048  8085  8094  8095  8103  8105  8118  8155
                    8164  8165  8173  8175  8188  8225  8234  8235  8243  8245  8257  8294  8303
                    8304  8312  8314  8327  8364  8373  8374  8382  8384  8439  8525  8533
F$HARD= 000004       1368#  8406  8437
F$HW  = 000013       1368#  1598  1616
F$INIT= 000006       1368#  3042  3174
F$JMP = 000050       1368#  3022  5682  5734  5784  6081  6208  6277  6346  6415  6484  6553  6622
                     6691   6760  6831  6901  6972  7042  7112  7182  7253  7323  7394  7464  7534
                     7604   7674  7744  7814  7884  7954  8025  8095  8165  8235  8304  8374
F$MOD = 000000       1368#  1374  8533
F$MSG = 000011       1368#  2492  2506  2509  2523  2526  2540  2543  2558  2561  2576  2579  2594
                     2597   2612  2615  2630  2633  2647  2650  2664  2667  2674  2677  2691  2694
                     2708   2711  2718  2721  2728  2731  2745  2748  2755  2758  2765  2768  2782
                     2785   2792  2795  2809  2812  2826  2829  2844  2847  2861  2864  2879  2882
                     2896   2899  2906  2909  2916  2919  2933  2936  2950  2953  2967  2970  2984
                     2988   3004
F$PROT= 000021       1368#  1490  1495
F$PWR = 000017       1368#
F$RPT = 000012       1368#  3018  3027
F$SEG = 000003       1368#  3359  3378  3383  3401  3422  3452  3477  3493  3496  3510  3530  3546
                     3549   3560  3580  3596  3599  3610  3631  3662  3690  3708  3712  3732  3754
                     3772   3776  3797  3819  3840  3843  3861  3883  3899  3903  3917  3982  4006
                     4011   4037  4060  4084  4089  4115  4138  4162  4167  4193  4216  4240  4245
                     4271   4294  4323  4328  4359  4382  4411  4416  4447  4470  4494  4499  4525
                     4548   4572  4577  4603  4626  4650  4655  4681  4704  4728  4733  4759  4782
                     4806   4811  4837  4860  4884  4889  4915  4938  4962  4967  4993  5016  5040
                     5045   5071  5094  5127  5132  5156  5179  5203  5208  5237  5263  5294  5298
                     5332   5362  5395  5400  5427  6112  6139  6173  6205  6242  6274  6311  6343
                     6380   6412  6449  6481  6518  6550  6587  6619  6656  6688  6725  6757  6796
                     6828   6866  6898  6937  6969  7007  7039  7077  7109  7147  7179  7218  7250
                     7288   7320  7359  7391  7429  7461  7499  7531  7569  7601  7639  7671  7709
                     7741   7779  7811  7849  7881  7919  7951  7990  8022  8060  8092  8130  8162
                     8200   8232  8269  8301  8339  8371
F$SOFT= 000005       1368#  8518  8523
F$SRV = 000010       1368#
```

```
F$SUB = 000002          1368#  5261  5339
F$SW  = 000014          1368#  1630  1636
F$TEST= 000001          1368#  3276  3303  3314  3327  3352  3404  3417  3455  3474  3514  3527  3564
                        3577   3614  3626  3665  3684  3735  3748  3800  3813  3864  3877  3920  3937
                        3961   3974  4040  4053  4118  4131  4196  4209  4274  4287  4362  4375  4450
                        4463   4528  4541  4606  4619  4684  4697  4762  4775  4840  4853  4918  4931
                        4996   5009  5074  5087  5159  5172  5240  5253  5342  5355  5430  5442  5476
                        5488   5521  5534  5576  5589  5635  5647  5687  5699  5739  5751  5789  5802
                        5853   5870  5910  5923  5972  5985  6018  6031  6087  6103  6142  6162  6216
                        6231   6285  6300  6354  6369  6423  6438  6492  6507  6561  6576  6630  6645
                        6699   6714  6769  6785  6839  6855  6910  6926  6980  6996  7050  7066  7120
                        7136   7190  7207  7261  7277  7331  7348  7402  7418  7472  7488  7542  7558
                        7612   7628  7682  7698  7752  7768  7822  7838  7892  7908  7962  7979  8033
                        8049   8103  8119  8173  8189  8243  8258  8312  8328  8382
GETPRM  010712          3079   3087#  3096
G$CNTO= 000200          1368#
G$DELM= 000372          1368#
G$DISP= 000003          1368#
G$EXCP= 000400          1368#
G$HILI= 000002          1368#
G$LOLI= 000001          1368#
G$NO  = 000000          1368#
G$OFFS= 000400          1368#  8410  8416  8421  8426
G$OFSI= 000376          1368#  8410  8416  8421  8426
G$PRMA= 000001          1368#  8416  8421
G$PRMD= 000002          1368#  8410  8426
G$PRML= 000000          1368#
G$RADA= 000140          1368#
G$RADB= 000000          1368#
G$RADD= 000040          1368#
G$RADL= 000120          1368#
G$RADO= 000020          1368#  8410  8416  8421  8426
G$XFER= 000004          1368#
G$YES = 000010          1368#  8410  8416  8421  8426
HELP  = 000000          1359#  1399  1488  1584  1914  3020  3025
HOE   = 100000 G        1730#
IBE   = 010000 G        1727#
IDU   = 000040 G        1720#
IER   = 020000 G        1728#
INIFLG  002674          1834#
INSTU   003536          2101   2112#
ISR   = 000100 G        1721#
IXE   = 004000 G        1726#
I$AU  = 000041          1368#  3252#  3256#
I$AUTO= 000041          1368#  3179#  3202#
I$CLN = 000041          1368#  3211#  3218#
I$DU  = 000041          1368#  3231#  3238#
I$HRD = 000041          8406#  8439#
I$INIT= 000041          1368#  3042#  3176#
I$MOD = 000041          1368#  1374#  8533#
I$MSG = 000041          1368#  2492#  2508#  2509#  2525#  2526#  2542#  2543#  2560#  2561#  2578#  2579#  2596#
                        2597#  2614#  2615#  2632#  2633#  2649#  2650#  2666#  2667#  2676#  2677#  2693#  2694#
                        2710#  2711#  2720#  2721#  2730#  2731#  2747#  2748#  2757#  2758#  2767#  2768#  2784#
                        2785#  2794#  2795#  2811#  2812#  2828#  2829#  2846#  2847#  2863#  2864#  2881#  2882#
                        2898#  2899#  2908#  2909#  2918#  2919#  2935#  2936#  2952#  2953#  2969#  2970#  2986#
                        2988#  3006#
```

```
I$PROT= 000040   1368#  1490#
I$PTAB= 000041   1368#
I$PWR = 000041   1368#
I$RPT = 000041   1368#  3018#  3029#
I$SEG = 000041   1368#  3275   3313   3351   3359#  3371   3380#  3383#  3394   3403#  3416   3422#  3435
                 3447   3454#  3473   3477#  3490   3495#  3496#  3507   3512#  3526   3530#  3543   3548#
                 3549#  3562#  3576   3580#  3593   3598#  3599#  3612#  3625   3631#  3645   3657   3664#
                 3683   3690#  3702   3710#  3712#  3725   3734#  3747   3754#  3766   3774#  3776#  3790
                 3799#  3812   3819#  3831   3842#  3843#  3854   3863#  3876   3883#  3896   3901#  3903#
                 3914   3919#  3936   3973   3982#  4000   4008#  4011#  4030   4039#  4052   4060#  4078
                 4086#  4089#  4108   4117#  4130   4138#  4156   4164#  4167#  4186   4195#  4208   4216#
                 4234   4242#  4245#  4264   4273#  4286   4294#  4316   4325#  4328#  4351   4361#  4374
                 4382#  4404   4413#  4416#  4439   444[illegible]  4462   4470#  4488   4496#  4499#  4518   4527#
                 4540   4548#  4566   4574#  4577#  4596   4605#  4618   4626#  4644   4652#  4655#  4674
                 4683#  4696   4704#  4722   4730#  4733#  4752   4761#  4774   4782#  4800   4808#  4811#
                 4830   4839#  4852   4860#  4878   4886#  4889#  4908   4917#  4930   4938#  4956   4964#
                 4967#  4986   4995#  5008   5016#  5034   5042#  5045#  5064   5073#  5086   5094#  5120
                 5129#  5132#  5148   5158#  5171   5179#  5197   5205#  5208#  5228   5239#  5252   5260
                 5263.  5288   5296#  5298#  5334#  5354   5362#  5388   5397#  5400#  5420   5429#  5441
                 5487   5533   5588   5646   5698   5750   5801   5869   5922   5984   6030   6102   6112#
                 6136   6141#  6161   6173#  6198   6207#  6230   6242#  6267   6276#  6299   6311#  6336
                 6345#  6368   6380#  6405   6414#  6437   6449#  6474   6483#  6506   6518#  6543   6552#
                 6575   6587#  6612   6621#  6644   6656#  6681   6690#  6713   6725#  6750   6759#  6784
                 6796#  6821   6830#  6854   6866#  6891   6900#  6925   6937#  6962   6971#  6995   7007#
                 7032   7041#  7065   7077#  7102   7111#  7135   7147#  7172   7181#  7206   7218#  7243
                 7252#  7276   7288#  7313   7322#  7347   7359#  7384   7393#  7417   7429#  7454   7463#
                 7487   7499#  7524   7533#  7557   7569#  7594   7603#  7627   7639#  7664   7673#  7697
                 7709#  7734   7743#  7767   7779#  7804   7813#  7837   7849#  7874   7883#  7907   7919#
                 7944   7953#  7978   7990#  8015   8024#  8048   8060#  8085   8094#  8118   8130#  8155
                 8164#  8188   8200#  8225   8234#  8257   8269#  8294   8303#  8327   8339#  8364   8373#
I$SETU= 000041   1368#
I$SFT = 000041   8518#  8525#
I$SRV = 000041   1368#
I$SUB = 000041   1368#  3275   3313   3351   3416   3473   3526   3576   3625   3683   3747   3812   3876
                 3936   3973   4052   4130   4208   4286   4374   4462   4540   4618   4696   4774   4852
                 4930   5008   5086   5171   5252   5260#  5339#  5341#  5354   5441   5487   5533   5588
                 5646   5698   5750   5801   5869   5922   5984   6030   6102   6161   6230   6299   6368
                 6437   6506   6575   6644   6713   6784   6854   6925   6995   7065   7135   7206   7276
                 7347   7417   7487   7557   7627   7697   7767   7837   7907   7978   8048   8118   8188
                 8257   8327
I$TST = 000041   1368#  3275#  3283   3300   3303#  3305#  3313#  3327#  3329#  3351#  3404#  3406#  3416#
                 3455#  3457#  3473#  3514#  3516#  3526#  3564#  3566#  3576#  3614#  3616#  3625#  3665#
                 3667#  3683#  3735#  3737#  3747#  3800#  3802#  3812#  3864#  3866#  3876#  3920#  3922#
                 3936#  3957   3961#  3963#  3973#  4040#  4042#  4052#  4118#  4120#  4130#  4196#  4198#
                 4208#  4274#  4276#  4286#  4362#  4364#  4374#  4450#  4452#  4462#  4528#  4530#  4540#
                 4606#  4608#  4618#  4684#  4686#  4696#  4762#  4764#  4774#  4840#  4842#  4852#  4918#
                 4920#  4930#  4996#  4998#  5008#  5074#  5076#  5086#  5159#  5161#  5171#  5240#  5242#
                 5252#  5260   5325   5342#  5344#  5354#  5430#  5432#  5441#  5476#  5478#  5487#  5521#
                 5523#  5533#  5576#  5578#  5588#  5635#  5637#  5646#  5679   5682   5687#  5689#  5698#
                 5731   5734   5739#  5741#  5750#  5781   5784   5789#  5791#  5801#  5853#  5855#  5869#
                 5910#  5912#  5922#  5972#  5974#  5984#  6018#  6020#  6030#  6065   6081   6087#  6089#
                 6102#  6142#  6144#  6161#  6208   6216#  6218#  6230#  6277   6285#  6287#  6299#  6346
                 6354#  6356#  6368#  6415   6423#  6425#  6437#  6484   6492#  6494#  6506#  6553   6561#
                 6563#  6575#  6622   6630#  6632#  6644#  6691   6699#  6701#  6713#  6760   6769#  6771#
                 6784#  6831   6839#  6841#  6854#  6901   6910#  6912#  6925#  6972   6980#  6982#  6995#
                 7042   7050#  7052#  7065#  7112   7120#  7122#  7135#  7182   7190#  7192#  7206#  7253
```

```
                        7261#  7263#  7276#  7323   7331#  7333#  7347#  7394   7402#  7404#  7417#  7464   7472#
                        7474#  7487#  7534   7542#  7544#  7557#  7604   7612#  7614#  7627#  7674   7682#  7684#
                        7697#  7744   7752#  7754#  7767#  7814   7822#  7824#  7837#  7884   7892#  7894#  7907#
                        7954   7962#  7964#  7978#  8025   8033#  8035#  8048#  8095   8103#  8105#  8118#  8165
                        8173#  8175#  8188#  8235   8243#  8245#  8257#  8304   8312#  8314#  8327#  8374   8382#
                        8384#
J$JMP = 000167          1368#  3022
KMACTV  002610          1801#
KMCSR   002716          1870#  3099*  3181   3276   3315   3353   3380   3423   3478   3531   3581   3814   3837
                        3881   3938   3980   4058   4136   4214   4292   4380   4468   4546   4624   4702   4780
                        4858   4936   5014   5092   5177   5254   5360   5443   5489   5535   5590   5651   5700
                        5752   5805   5871   5924   5986   6032   6104   6163   6232   6301   6370   6439   6508
                        6577   6646   6715   6786   6856   6927   6997   7067   7137   7208   7278   7349   7419
                        7489   7559   7629   7699   7769   7839   7909   7980   8050   8120   8190   8259   8329
KMCSRH  002720          1871#  1991*  1992*  2006*  2008*  2009*  3101*  3102*
KMCTL   002722          1872#  3104*  3105*  3633
KMNUM   002612          1802#
KMPO4   002724          1873#  2163*  2180   2202*  3107*  3108*  3687
KMPO6   002726          1874#  2007*  3110*  3111*  3751   5806   5807
KMRLVL  002710          1867#  2124*  3115*  3116*
KMRVEC  002706          1866#  2122*  3113*
KMTLVL  002714          1869#  2125*  3121*  3122*
KMTVEC  002712          1868#  2123*  3118*  3119*
LNUNIT  035126          8467#
LOCK    002442          1771#
LOE   = 040000 G        1729#
LOGDEV  002552          1785#  3082*  3088*  3089   3092   3194
LOKFLG  002676          1836#
LOOPBK  035323          8489#
LOT   = 000010 G        1718#
LTN.ED= 000114          8545#
L$ACP   002110 G        1476#
L$APT   002036 G        1434#
L$AU    011364 G        1461   3252#
L$AUT   002070 G        1460#
L$AUTO  011256 G        1477   3179#
L$CCP   002106 G        1474#
L$CLEA  011354 G        1475   3211#
L$CO    002032 G        1430#
L$DEPO  002011 G        1412#
L$DESC  002414 G        1467   1760#
L$DESP  002076 G        1466#
L$DEVP  002060 G        1452#
L$DISP  002132 G        1437   1505#
L$DLY   002116 G        1482#
L$DTP   002040 G        1436#
L$DTYP  002034 G        1432#
L$DU    011360 G        1463   3231#
L$DUT   002072 G        1462#
L$DVTY  003130 G        1453   1902#
L$EF    002052 G        1447#
L$ENVI  002044 G        1440#
L$ETP   002102 G        1470#
L$EXP1  002046 G        1442#
L$EXP4  002064 G        1456#
L$EXP5  002066 G        1458#
```

```
L$HARD  034640 G        1419    8406    8407#
L$HIME  002120 G        1484#
L$HPCP  002016 G        1418#
L$HPTP  002022 G        1422#
L$HW    002364 G        1423    1598    1599#
L$ICP   002104 G        1472#
L$INIT  010570 G        1473    3042#
L$LADP  002026 G        1426#
L$LAST  040004 G        1427    8544#
L$LOAD  002100 G        1468#
L$LUN   002074 G        1464#
L$MREV  002050 G        1444#
L$NAME  002000 G        1401#
L$PRIO  002042 G        1438#
L$PROT  002122 G        1479    1490#
L$PRT   002112 G        1478#
L$REPP  002062 G        1454#
L$REV   002010 G        1410#
L$RPT   010562 G        3018#
L$SOFT  035404 G        8518    8519#
L$SPC   002056 G        1450#
L$SPCP  002020 G        1420#
L$SPTP  002024 G        1424#
L$STA   002030 G        1428#
L$SW    002414 G        1630    1631#
L$TEST  002114 G        1480#
L$TIML  002014 G        1416#
L$UNIT  002012 G        1414#   3089
L10001  002412          1598    1616#
L10002  002414          1630    1636#
L10003  006130          2506#
L10004  006206          2523#
L10005  006264          2540#
L10006  006346          2558#
L10007  006430          2576#
L10010  006512          2594#
L10011  006574          2612#
L10012  006656          2630#
L10013  006734          2647#
L10014  007012          2664#
L10015  007040          2674#
L10016  007116          2691#
L10017  007174          2708#
L10020  007222          2718#
L10021  007250          2728#
L10022  007326          2745#
L10023  007354          2755#
L10024  007402          2765#
L10025  007460          2782#
L10026  007506          2792#
L10027  007564          2809#
L10030  007642          2826#
L10031  007724          2844#
L10032  010002          2861#
L10033  010064          2879#
L10034  010142          2896#
```

L10035	010170	2506#		
L10036	010216	2916#		
L10037	010274	2933#		
L10040	010352	2950#		
L10041	010430	2967#		
L10042	010506	2984#		
L10043	010560	3004#		
L10044	010566	3023	3027#	
L10045	011254	3174#		
L10046	011352	3200#		
L10047	011356	3216#		
L10050	011362	3236#		
L10051	011364	3254#		
L10052	011512	3284	3301	3303#
L10053	011556	3327#		
L10054	011744	3404#		
L10055	012110	3455#		
L10056	012244	3514#		
L10057	012374	3564#		
L10060	012524	3614#		
L10061	012666	3665#		
L10062	013052	3735#		
L10063	013236	3800#		
L10064	013424	3864#		
L10065	013574	3920#		
L10066	013702	3958	3961#	
L10067	014132	4040#		
L10070	014362	4118#		
L10071	014612	4196#		
L10072	015042	4274#		
L10073	015336	4362#		
L10074	015632	4450#		
L10075	016062	4528#		
L10076	016312	4606#		
L10077	016542	4684#		
L10100	016772	4762#		
L10101	017222	4840#		
L10102	017452	4918#		
L10103	017702	4996#		
L10104	020132	5074#		
L10105	020430	5159#		
L10106	020660	5240#		
L10107	021224	5[illegible]	5342#	
L10110	021222	5[illegible]9#		
L10111	021536	5430#		
L10112	021700	5476#		
L10113	022042	5521#		
L10114	022216	5576#		
L10115	022422	5635#		
L10116	022566	5680	5683	5687#
L10117	022736	5732	5735	5739#
L10120	023102	5782	5785	5789#
L10121	023270	5853#		
L10122	023504	5910#		
L10123	023722	5972#		
L10124	024054	6018#		

```
L10125  024272          6066    6082    6087#
L10126  024422          6142#
L10127  024636          6209    6216#
L10130  025042          6278    6285#
L10131  025246          6347    6354#
L10132  025452          6416    6423#
L10133  025656          6485    6492#
L10134  026062          6554    6561#
L10135  026266          6623    6630#
L10136  026472          6692    6699#
L10137  026700          6761    6769#
L10140  027104          6832    6839#
L10141  027310          6902    6910#
L10142  027514          6973    6980#
L10143  027720          7043    7050#
L10144  030124          7113    7120#
L10145  030330          7183    7190#
L10146  030534          7254    7261#
L10147  030740          7324    7331#
L10150  031144          7395    7402#
L10151  031350          7465    7472#
L10152  031554          7535    7542#
L10153  031760          7605    7612#
L10154  032164          7675    7682#
L10155  032370          7745    7752#
L10156  032574          7815    7822#
L10157  033000          7885    7892#
L10160  033204          7955    7962#
L10161  033410          8026    8033#
L10162  033614          8096    8103#
L10163  034020          8166    8173#
L10164  034224          8236    8243#
L10165  034430          8305    8312#
L10166  034634          8375    8382#
L10167  034704          8406    8438#
L10170  035404          8518    8524#
MASKX   002572          1793#
MDATA   002506          1778#
MEMDAT  002654          1823#   6169    6238    6307    6376    6445    6514    6583    6652    6721    6792    6862    6933
                        7003    7073    7143    7214    7284    7355    7425    7495    7565    7635    7705    7775    7845
                        7915    7986    8056    8126    8196    8265    8335
MEMEND  037776          1798    8539#
MEMLD   003640          2148#   6107    6168    6237    6306    6375    6444    6513    6582    6651    6720    6791    6861
                        6932    7002    7072    7142    7213    7283    7354    7424    7494    7564    7634    7704    7774
                        7844    7914    7985    8055    8125    8195    8264    8334
MEMLD2  003650          2155#
MEMLIM  002604          1798#   6074
MEMSET  003424          2082#
MEMSZ   002606          1799#   3160*   3168*
MRO     002624          1807#   2184    2838    2873    3292    3321    3366    3389    3419*   3430    3442    3485    3502
                        3538    3555    3588    3605    3628*   3640    3652    3684*   3697    3720    3748*   3761    3785
                        3826    3849    3891    3909    3951    3976*   4055*   4133*   4211*   4289*   4377*   4465*   4543*
                        4621*   4699*   4777*   4855*   4933*   5011*   5463    5469    5508    5514    5568    5618    5626
                        5674    5726    5776    5847    5901    5950    5965    6012    6060    6131    6193    6262    6331
                        6400    6469    6538    6607    6676    6745    6816    6886    6957    7027    7097    7167    7238
                        7308    7379    7449    7519    7589    7659    7729    7799    7869    7939    8010    8080    8150
```

```
                        8220    8289    8359
NEWST   010704          3060    3066    3080#   3090
NEXT    002440          1770#
NPRSET  003574          2129#   5654    5704    5756    5808    5879    5927    5989    6044
ONE     002602          1797#
O$APTS= 000000          1368#   1428
O$AU  = 000001          1368#   1398#   1460
O$BGNR= 000000          1368#   1454
O$BGNS= 000000          1368#   1420
O$DU  = 000001          1368#   1398#   1462
O$ERRT= 000000          1368#   1470
O$GNSW= 000000          1368#   1424
O$POIN= 000001          1368#   1398#   1486
O$SETU= 000000          1368#   1414    8542
PNT   = 001000 G        1724#
POP2SP= 022626          1739#
PRI   = 002000 G        1725#
PRIRTY  035064          8427    8461#
PRI00 = 000000 G        1713#   5459    5504
PRI01 = 000040 G        1712#
PRI02 = 000100 G        1711#
PRI03 = 000140 G        1710#
PRI04 = 000200 G        1709#
PRI05 = 000240 G        1708#
PRI06 = 000300 G        1707#
PRI07 = 000340 G        1706#   5452    5497
PSTACK  002554          1786#   3047*
QV.FLG  002677          1837#
RAMDAT  003370          2065#
REGADR  002730          1878#
RETADR  002563          1789#
RUN     002622          1806#
SAVACT  002614          1803#
SAVE4   002650          1817#   3050*   3053    3197    3297    5908
SAVE6   002652          1818#   3051*   3054    3198    3298    5907
SAVNUM  002616          1804#
SAVPC   002576          1795#
SAVSP   002574          1794#
SETBR0  003320          2025#
SETBR1  003330          2032#
SETBR4  003340          2040#
SETBR7  003350          2048#
SETC    004076          2226#   7289    7360    7430    7500    7570    7640    7710    7780    7850    7920    7991    8061
                        8131    8201    8270    8340
SETVEC  003552          2119#   5447    5492    5549    5605
SETZ    003360          2057#
SFPTBL  002414 G        1632#
SPDAT   002664          1826#   6171    6240    6309    6378    6447    6516    6585    6654    6723    6794    6864    6935
                        7005    7075    7145    7216    7286    7357    7427    7497    7567    7637    7707    7777    7847
                        7917    7988    8058    8128    8198    8267    8337
SPLD    004012          2195#   6109    6170    6239    6308    6377    6446    6515    6584    6653    6722    6793    6863
                        6934    7004    7074    7144    7215    7285    7356    7426    7496    7566    7636    7706    7776
                        7846    7916    7987    8057    8127    8197    8266    8336
SSTACK  003130          1882#   3045
STAT    002566          1791#
STAT1   002700          1859#   3124*   3126*   3130*   3141*   3145*   3148*   5542    5597
```

```
STAT2   002702          1660#  3151*  3153*
STAT3   002704          1861#
STOP    023302          5874#
STRTSW  002564          1790#
SUBRPC  002556          1787#
SVCGBL= 000000          1368#  1374   1381#  1401   1402   1410   1411   1412   1413   1414   1415   1416   1417
                        1418   1419   1420   1421   1422   1423   1424   1425   1426   1427   1428   1429   1430
                        1431   1432   1433   1434   1435   1436   1437   1438   1439   1440   1441   1442   1443
                        1444   1445   1447   1448   1450   1451   1452   1453   1454   1455   1456   1457   1458
                        1459   1460   1461   1462   1463   1464   1465   1466   1467   1468   1469   1470   1471
                        1472   1473   1474   1475   1476   1477   1478   1479   1480   1481   1482   1483   1484
                        1485   1490   1491   1505   1506   1599   1600   1601   1631   1632   1633   1760   1761
                        1902   1903   2492   2493   2509   2510   2526   2527   2543   2544   2561   2562   2579
                        2580   2597   2598   2615   2616   2633   2634   2650   2651   2667   2668   2677   2678
                        2694   2695   2711   2712   2721   2722   2731   2732   2748   2749   2758   2759   2768
                        2769   2785   2786   2795   2796   2812   2813   2829   2830   2847   2848   2864   2865
                        2882   2883   2899   2900   2909   2910   2919   2920   2936   2937   2953   2954   2970
                        2971   2988   2989   3018   3019   3042   3043   3179   3180   3211   3212   3231   3232
                        3252   3253   8407   8408   8519   8520   8544#  8545
SVCINS= 000000          1368#  1378#  1402   1403   1404   1405   1406   1407   1408   1409   1410   1411   1412
                        1413   1414   1415   1416   1417   1418   1419   1420   1421   1422   1423   1424   1425
                        1426   1427   1428   1429   1430   1431   1432   1433   1434   1435   1436   1437   1438
                        1439   1440   1441   1442   1443   1444   1445   1446   1447   1448   1449   1450   1451
                        1452   1453   1454   1455   1456   1457   1458   1459   1460   1461   1462   1463   1464
                        1465   1466   1467   1468   1469   1470   1471   1472   1473   1474   1475   1476   1477
                        1478   1479   1480   1481   1482   1483   1484   1485   1486   1504   1505   1506   1507
                        1508   1509   1510   1511   1512   1513   1514   1515   1516   1517   1518   1519   1520
                        1521   1522   1523   1524   1525   1526   1527   1528   1529   1530   1531   1532   1533
                        1534   1535   1536   1537   1538   1539   1540   1541   1542   1543   1544   1545   1546
                        1547   1548   1 49   1550   1551   1552   1553   1554   1555   1556   1557   1558   1559
                        1560   1561   1562   1563   1564   1565   1566   1567   1568   1569   1570   1571   1572
                        1573   1574   1575   1576   1577   1578   1579   1580   1581   1582   1598   1599   1630
                        1631   1761   1765   1766   1903   1907   1908   2184   2185   2186   2187   2188   2493
                        2494   2495   2496   2497   2498   2499   2500   2501   2502   2503   2504   2505   2506
                        2507   2508   2510   2511   2512   2513   2514   2515   2516   2517   2518   2519   2520
                        2521   2522   2523   2524   2525   2527   2528   2529   2530   2531   2532   2533   2534
                        2535   2536   2537   2538   2539   2540   2541   2542   2544   2545   2546   2547   2548
                        2549   2550   2551   2552   2553   2554   2555   2556   2557   2558   2559   2560   2562
                        2563   2564   2565   2566   2567   2568   2569   2570   2571   2572   2573   2574   2575
                        2576   2577   2578   2580   2581   2582   2583   2584   2585   2586   2587   2588   2589
                        2590   2591   2592   2593   2594   2595   2596   2598   2599   2600   2601   2602   2603
                        2604   2605   2606   2607   2608   2609   2610   2611   2612   2613   2614   2616   2617
                        2618   2619   2620   2621   2622   2623   2624   2625   2626   2627   2628   2629   2630
                        2631   2632   2634   2635   2636   2637   2638   2639   2640   2641   2642   2643   2644
                        2645   2646   2647   2648   2649   2651   2652   2653   2654   2655   2656   2657   2658
                        2659   2660   2661   2662   2663   2664   2665   2666   2668   2669   2670   2671   2672
                        2673   2674   2675   2676   2678   2679   2680   2681   2682   2683   2684   2685   2686
                        2687   2688   2689   2690   2691   2692   2693   2695   2696   2697   2698   2699   2700
                        2701   2702   2703   2704   2705   2706   2707   2708   2709   2710   2712   2713   2714
                        2715   2716   2717   2718   2719   2720   2722   2723   2724   2725   2726   2727   2728
                        2729   2730   2732   2733   2734   2735   2736   2737   2738   2739   2740   2741   2742
                        2743   2744   2745   2746   2747   2749   2750   2751   2752   2753   2754   2755   2756
                        2757   2759   2760   2761   2762   2763   2764   2765   2766   2767   2769   2770   2771
                        2772   2773   2774   2775   2776   2777   2778   2779   2780   2781   2782   2783   2784
                        2786   2787   2788   2789   2790   2791   2792   2793   2794   2796   2797   2798   2799
                        2800   2801   2802   2803   2804   2805   2806   2807   2808   2809   2810   2811   2813
```

2814	2815	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825	2826
2827	2828	2830	2831	2832	2833	2834	2835	2836	2837	2838	2839	2840
2841	2842	2843	2844	2845	2846	2848	2849	2850	2851	2852	2853	2854
2855	2856	2857	2858	2859	2860	2861	2862	2863	2865	2866	2867	2868
2869	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879	2880	2881
2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893	2894	2895
2896	2897	2898	2900	2901	2902	2903	2904	2905	2906	2907	2908	2910
2911	2912	2913	2914	2915	2916	2917	2918	2920	2921	2922	2923	2924
2925	2926	2927	2928	2929	2930	2931	2932	2933	2934	2935	2937	2938
2939	2940	2941	2942	2943	2944	2945	2946	2947	2948	2949	2950	2951
2952	2954	2955	2956	2957	2958	2959	2960	2961	2962	2963	2964	2965
2966	2967	2968	2969	2971	2972	2973	2974	2975	2976	2977	2978	2979
2980	2981	2982	2983	2984	2985	2986	2990	2991	2992	2993	2994	2995
2996	2997	2998	2999	3000	3001	3002	3003	3005	3006	3022	3023	3024
3028	3029	3057	3058	3059	3060	3061	3063	3064	3065	3066	3067	3069
3070	3071	3072	3073	3076	3077	3078	3079	3080	3092	3093	3094	3095
3096	3097	3175	3176	3194	3195	3196	3201	3202	3213	3214	3217	3218
3234	3235	3237	3238	3255	3256	3283	3284	3285	3292	3293	3294	3295
3296	3300	3301	3302	3304	3305	3321	3322	3323	3324	3325	3328	3329
3359	3360	3366	3367	3368	3369	3370	3371	3372	3373	3379	3380	3383
3384	3389	3390	3391	3392	3393	3394	3395	3396	3402	3403	3405	3406
3422	3423	3430	3431	3432	3433	3434	3435	3436	3437	3442	3443	3444
3445	3446	3447	3448	3449	3453	3454	3456	3457	3477	3478	3485	3486
3487	3488	3489	3490	3491	3492	3494	3495	3496	3497	3502	3503	3504
3505	3506	3507	3508	3509	3511	3512	3515	3516	3530	3531	3538	3539
3540	3541	3542	3543	3544	3545	3547	3548	3549	3550	3555	3556	3557
3558	3559	3561	3562	3565	3566	3580	3581	3588	3589	3590	3591	3592
3593	3594	3595	3597	3598	3599	3600	3605	3606	3607	3608	3609	3611
3612	3615	3616	3631	3632	3640	3641	3642	3643	3644	3645	3646	3647
3652	3653	3654	3655	3656	3657	3658	3659	3663	3664	3666	3667	3690
3691	3697	3698	3699	3700	3701	3702	3703	3704	3709	3710	3712	3713
3720	3721	3722	3723	3724	3725	3726	3727	3733	3734	3736	3737	3754
3755	3761	3762	3763	3764	3765	3766	3767	3768	3773	3774	3776	3777
3785	3786	3787	3788	3789	3790	3791	3792	3798	3799	3801	3802	3819
3820	3826	3827	3828	3829	3830	3831	3832	3833	3841	3842	3843	3844
3849	3850	3851	3852	3853	3854	3855	3856	3862	3863	3865	3866	3883
3884	3891	3892	3893	3894	3895	3896	3897	3898	3900	3901	3903	3904
3909	3910	3911	3912	3913	3914	3915	3916	3918	3919	3921	3922	3951
3952	3953	3954	3955	3957	3958	3959	3962	3963	3982	3983	3995	3996
3997	3998	3999	4000	4001	4002	4007	4008	4011	4012	4025	4026	4027
4028	4029	4030	4031	4032	4038	4039	4041	4042	4060	4061	4073	4074
4075	4076	4077	4078	4079	4080	4085	4086	4089	4090	4103	4104	4105
4106	4107	4108	4109	4110	4116	4117	4119	4120	4138	4139	4151	4152
4153	4154	4155	4156	4157	4158	4163	4164	4167	4168	4181	4182	4183
4184	4185	4186	4187	4188	4194	4195	4197	4198	4216	4217	4229	4230
4231	4232	4233	4234	4235	4236	4241	4242	4245	4246	4259	4260	4261
4262	4263	4264	4265	4266	4272	4273	4275	4276	4294	4295	4311	4312
4313	4314	4315	4316	4317	4318	4324	4325	4328	4329	4346	4347	4348
4349	4350	4351	4352	4353	4360	4361	4363	4364	4382	4383	4399	4400
4401	4402	4403	4404	4405	4406	4412	4413	4416	4417	4434	4435	4436
4437	4438	4439	4440	4441	4448	4449	4451	4452	4470	4471	4483	4484
4485	4486	4487	4488	4489	4490	4495	4496	4499	4500	4513	4514	4515
4516	4517	4518	4519	4520	4526	4527	4529	4530	4548	4549	4561	4562
4563	4564	4565	4566	4567	4568	4573	4574	4577	4578	4591	4592	4593
4594	4595	4596	4597	4598	4604	4605	4607	4608	4626	4627	4639	4640
4641	4642	4643	4644	4645	4646	4651	4652	4655	4656	4669	4670	4671

4672	4673	4674	4675	4676	4682	4683	4685	4686	4704	4705	4717	4718
4719	4720	4721	4722	4723	4724	4729	4730	4733	4734	4747	4748	4749
4750	4751	4752	4753	4754	4760	4761	4763	4764	4782	4783	4795	4796
4797	4798	4799	4800	4801	4802	4807	4808	4811	4812	4825	4826	4827
4828	4829	4830	4831	4832	4838	4839	4841	4842	4860	4861	4873	4874
4875	4876	4877	4878	4879	4880	4885	4886	4889	4890	4903	4904	4905
4906	4907	4908	4909	4910	4916	4917	4919	4920	4938	4939	4951	4952
4953	4954	4955	4956	4957	4958	4963	4964	4967	4968	4981	4982	4983
4984	4985	4986	4987	4988	4994	4995	4997	4998	5016	5017	5029	5030
5031	5032	5033	5034	5035	5036	5041	5042	5045	5046	5059	5060	5061
5062	5063	5064	5065	5066	5072	5073	5075	5076	5094	5095	5115	5116
5117	5118	5119	5120	5121	5122	5128	5129	5132	5133	5143	5144	5145
5146	5147	5148	5149	5150	5157	5158	5160	5161	5179	5180	5192	5193
5194	5195	5196	5197	5198	5199	5204	5205	5208	5209	5223	5224	5225
5226	5227	5228	5229	5230	5238	5239	5241	5242	5261	5262	5263	5264
5283	5284	5285	5286	5287	5288	5289	5290	5295	5296	5298	5299	5320
5321	5322	5323	5324	5325	5326	5327	5333	5334	5340	5341	5343	5344
5362	5363	5383	5384	5385	5386	5387	5388	5389	5390	5396	5397	5400
5401	5415	5416	5417	5418	5419	5420	5421	5422	5428	5429	5431	5432
5445	5446	5452	5453	5454	5459	5460	5461	5463	5464	5465	5466	5467
5469	5470	5471	5472	5473	5477	5478	5497	5498	5499	5504	5505	5506
5508	5509	5510	5511	5512	5514	5515	5516	5517	5518	5522	5523	5540
5541	5542	5558	5559	5560	5566	5567	5568	5569	5570	5571	5572	5577
5578	5595	5596	5597	5614	5615	5616	5618	5619	5620	5621	5622	5626
5627	5628	5629	5630	5636	5637	5648	5649	5674	5675	5676	5677	5678
5679	5680	5681	5682	5683	5684	5688	5689	5726	5727	5728	5729	5730
5731	5732	5733	5734	5735	5736	5740	5741	5776	5777	5778	5779	5780
5781	5782	5783	5784	5785	5786	5790	5791	5847	5848	5849	5850	5851
5854	5855	5901	5902	5903	5904	5905	5911	5912	5950	5951	5952	5953
5954	5965	5966	5967	5968	5969	5973	5974	6012	6013	6014	6015	6016
6019	6020	6060	6061	6062	6063	6064	6065	6066	6067	6081	6082	6083
6088	6089	6112	6113	6131	6132	6133	6134	6135	6136	6137	6138	6140
6141	6143	6144	6173	6174	6193	6194	6195	6196	6197	6198	6199	6200
6206	6207	6208	6209	6210	6217	6218	6242	6243	6262	6263	6264	6265
6266	6267	6268	6269	6275	6276	6277	6278	6279	6286	6287	6311	6312
6331	6332	6333	6334	6335	6336	6337	6338	6344	6345	6346	6347	6348
6355	6356	6380	6381	6400	6401	6402	6403	6404	6405	6406	6407	6413
6414	6415	6416	6417	6424	6425	6449	6450	6469	6470	6471	6472	6473
6474	6475	6476	6482	6483	6484	6485	6486	6493	6494	6518	6519	6538
6539	6540	6541	6542	6543	6544	6545	6551	6552	6553	6554	6555	6562
6563	6587	6588	6607	6608	6609	6610	6611	6612	6613	6614	6620	6621
6622	6623	6624	6631	6632	6656	6657	6676	6677	6678	6679	6680	6681
6682	6683	6689	6690	6691	6692	6693	6700	6701	6725	6726	6745	6746
6747	6748	6749	6750	6751	6752	6758	6759	6760	6761	6762	6770	6771
6796	6797	6816	6817	6818	6819	6820	6821	6822	6823	6829	6830	6831
6832	6833	6840	6841	6866	6867	6886	6887	6888	6889	6890	6891	6892
6893	6899	6900	6901	6902	6903	6911	6912	6937	6938	6957	6958	6959
6960	6961	6962	6963	6964	6970	6971	6972	6973	6974	6981	6982	7007
7008	7027	7028	7029	7030	7031	7032	7033	7034	7040	7041	7042	7043
7044	7051	7052	7077	7078	7097	7098	7099	7100	7101	7102	7103	7104
7110	7111	7112	7113	7114	7121	7122	7147	7148	7167	7168	7169	7170
7171	7172	7173	7174	7180	7181	7182	7183	7184	7191	7192	7218	7219
7238	7239	7240	7241	7242	7243	7244	7245	7251	7252	7253	7254	7255
7262	7263	7288	7289	7308	7309	7310	7311	7312	7313	7314	7315	7321
7322	7323	7324	7325	7332	7333	7359	7360	7379	7380	7381	7382	7383
7384	7385	7386	7392	7393	7394	7395	7396	7403	7404	7429	7430	7449

```
                    7450  7451  7452  7453  7454  7455  7456  7462  7463  7464  7465  7466  7473
                    7474  7499  7500  7519  7520  7521  7522  7523  7524  7525  7526  7532  7533
                    7534  7535  7536  7543  7544  7569  7570  7589  7590  7591  7592  7593  7594
                    7595  7596  7602  7603  7604  7605  7606  7613  7614  7639  7640  7659  7660
                    7661  7662  7663  7664  7665  7666  7672  7673  7674  7675  7676  7683  7684
                    7709  7710  7729  7730  7731  7732  7733  7734  7735  7736  7742  7743  7744
                    7745  7746  7753  7754  7779  7780  7799  7800  7801  7802  7803  7804  7805
                    7806  7812  7813  7814  7815  7816  7823  7824  7849  7850  7869  7870  7871
                    7872  7873  7874  7875  7876  7882  7883  7884  7885  7886  7893  7894  7919
                    7920  7939  7940  7941  7942  7943  7944  7945  7946  7952  7953  7954  7955
                    7956  7963  7964  7990  7991  8010  8011  8012  8013  8014  8015  8016  8017
                    8023  8024  8025  8026  8027  8034  8035  8060  8061  8080  8081  8082  8083
                    8084  8085  8086  8087  8093  8094  8095  8096  8097  8104  8105  8130  8131
                    8150  8151  8152  8153  8154  8155  8156  8157  8163  8164  8165  8166  8167
                    8174  8175  8200  8201  8220  8221  8222  8223  8224  8225  8226  8227  8233
                    8234  8235  8236  8237  8244  8245  8269  8270  8289  8290  8291  8292  8293
                    8294  8295  8296  8302  8303  8304  8305  8306  8313  8314  8339  8340  8359
                    8360  8361  8362  8363  8364  8365  8366  8372  8373  8374  8375  8376  8383
                    8384  8406  8407  8410  8411  8412  8413  8414  8415  8416  8417  8418  8419
                    8420  8421  8422  8423  8424  8425  8426  8427  8428  8429  8430  8431  8437
                    8438  8518  8519  8523  8524  8541  8542  8543  8544
SVCSUB= 000000       1368#  1380#  5260  5261
SVCTAG= 000000       1368#  1382#  1616  1617  1636  1637  2506  2507  2523  2524  2540  2541  2558
                    2559  2576  2577  2594  2595  2612  2613  2630  2631  2647  2648  2664  2665
                    2674  2675  2691  2692  2708  2709  2718  2719  2728  2729  2745  2746  2755
                    2756  2765  2766  2782  2783  2792  2793  2809  2810  2826  2827  2844  2845
                    2861  2862  2879  2880  2896  2897  2906  2907  2916  2917  2933  2934  2950
                    2951  2967  2968  2984  2985  3004  3005  3027  3028  3174  3175  3200  3201
                    3216  3217  3236  3237  3254  3255  3303  3304  3327  3328  3378  3379  3401
                    3402  3404  3405  3452  3453  3455  3456  3493  3494  3510  3511  3514  3515
                    3546  3547  3560  3561  3564  3565  3594  3597  3610  3611  3614  3615  3662
                    3663  3665  3666  3708  3709  3732  3733  3735  3736  3772  3773  3797  3798
                    3800  3801  3840  3841  3861  3862  3864  3865  3899  3900  3917  3918  3920
                    3921  3961  3962  4006  4007  4037  4038  4040  4041  4084  4085  4115  4116
                    4118  4119  4162  4163  4193  4194  4196  4197  4240  4241  4271  4272  4274
                    4275  4323  4324  4359  4360  4362  4363  4411  4412  4447  4448  4450  4451
                    4494  4495  4525  4526  4528  4529  4572  4573  4603  4604  4606  4607  4650
                    4651  4681  4682  4684  4685  4728  4729  4759  4760  4762  4763  4806  4807
                    4837  4838  4840  4841  4884  4885  4915  4916  4918  4919  4962  4963  4993
                    4994  4996  4997  5040  5041  5071  5072  5074  5075  5127  5128  5156  5157
                    5159  5160  5203  5204  5237  5238  5240  5241  5294  5295  5332  5333  5339
                    5340  5342  5343  5395  5396  5427  5428  5430  5431  5476  5477  5521  5522
                    5576  5577  5635  5636  5687  5688  5739  5740  5789  5790  5853  5854  5910
                    5911  5972  5973  6018  6019  6087  6088  6139  6140  6142  6143  6205  6206
                    6216  6217  6274  6275  6285  6286  6343  6344  6354  6355  6412  6413  6423
                    6424  6481  6482  6492  6493  6550  6551  6561  6562  6619  6620  6630  6631
                    6688  6689  6699  6700  6757  6758  6769  6770  6828  6829  6839  6840  6898
                    6899  6910  6911  6969  6970  6980  6981  7039  7040  7050  7051  7109  7110
                    7120  7121  7179  7180  7190  7191  7250  7251  7261  7262  7320  7321  7331
                    7332  7391  7392  7402  7403  7461  7462  7472  7473  7531  7532  7542  7543
                    7601  7602  7612  7613  7671  7672  7682  7683  7741  7742  7752  7753  7811
                    7812  7822  7823  7881  7882  7892  7893  7951  7952  7962  7963  8022  8023
                    8033  8034  8092  8093  8103  8104  8162  8163  8173  8174  8232  8233  8243
                    8244  8301  8302  8312  8313  8371  8372  8382  8383  8438  8439  8524  8525
SVCTST= 000000       1368#  1379#  3275  3276  3313  3314  3351  3352  3416  3417  3473  3474  3526
                    3527  3576  3577  3625  3626  3683  3684  3747  3748  3812  3813  3876  3877
```

```
                    3536  3937  3973  3974  4052  4053  4130  4131  4208  4209  4286  4287  4374
                    4375  4462  4463  4540  4541  4618  4619  4696  4697  4774  4775  4852  4853
                    4930  4931  5008  5009  5086  5087  5171  5172  5252  5253  5354  5355  5441
                    5442  5487  5488  5533  5534  5588  5589  5646  5647  5698  5699  5750  5751
                    5801  5802  5869  5870  5922  5923  5984  5985  6030  6031  6102  6103  6161
                    6162  6230  6231  6299  6300  6368  6369  6437  6438  6506  6507  6575  6576
                    6644  6645  6713  6714  6784  6785  6854  6855  6925  6926  6995  6996  7065
                    7066  7135  7136  7206  7207  7276  7277  7347  7348  7417  7418  7487  7488
                    7557  7558  7627  7628  7697  7698  7767  7768  7837  7838  7907  7908  7978
                    7979  8048  8049  8118  8119  8188  8189  8257  8258  8327  8328
SWPAC1  035217      8477#
SWPAC2  035260      8483#
S$LSYM= 010000      1368#  1617#  1637#  2507#  2524#  2541#  2559#  2577#  2595#  2613#  2631#  2648#  2665#
                    2675#  2692#  2709#  2719#  2729#  2746#  2756#  2766#  2783#  2793#  2810#  2827#  2845#
                    2862#  2880#  2897#  2907#  2917#  2934#  2951#  2968#  2985#  3005#  3028#  3175#  3201#
                    3217#  3237#  3255#  3304#  3328#  3359#  3383#  3405#  3422#  3456#  3477#  3496#  3515#
                    3530#  3549#  3565#  3580#  3599#  3615#  3631#  3666#  3690#  3712#  3736#  3754#  3776#
                    3801#  3819#  3843#  3865#  3883#  3903#  3921#  3962#  3982#  4011#  4041#  4060#  4089#
                    4119#  4138#  4167#  4197#  4216#  4245#  4275#  4294#  4328#  4363#  4382#  4416#  4451#
                    4470#  4499#  4529#  4548#  4577#  4607#  4626#  4655#  4685#  4704#  4733#  4763#  4782#
                    4811#  4841#  4860#  4889#  4919#  4938#  4967#  4997#  5016#  5045#  5075#  5094#  5132#
                    5160#  5179#  5208#  5241#  5263#  5298#  5340#  5343#  5362#  5400#  5431#  5477#  5522#
                    5577#  5636#  5688#  5740#  5790#  5854#  5911#  5973#  6019#  6088#  6112#  6143#  6173#
                    6217#  6242#  6286#  6311#  6355#  6380#  6424#  6449#  6493#  6518#  6562#  6587#  6631#
                    6656#  6700#  6725#  6770#  6796#  6840#  6866#  6911#  6937#  6981#  7007#  7051#  7077#
                    7121#  7147#  7191#  7218#  7262#  7288#  7332#  7359#  7403#  7429#  7473#  7499#  7543#
                    7569#  7613#  7639#  7683#  7709#  7753#  7779#  7823#  7849#  7893#  7919#  7963#  7990#
                    8034#  8060#  8104#  8130#  8174#  8200#  8244#  8269#  8313#  8339#  8383#  8439#  8525#
TDATA   024424      6108   6110   6144#
TEMP    002444      1776#
TFM1    004123      2251#  2553   2571   2589   2607   2625
,TFM2   004146      2255#  2501   2518   2535   2642   2659   2686   2740   2804   2821   2856   2891   2928
                    2945   2962   2979
TFM27   004200      2261#  2839   2874
TFM37   004225      2265#  2998
TFM5    004163      2258#  2703   2777
TOUTP   023470      5899   5907#
TOUTT   023464      5874   5906#
TYPE    002630      1809#  3161*  3166*  3171*
T$ARGC= 000002      1402#  1403#  1404#  1405#  1406#  1407#  2493#  2498   2499#  2505   2510#  2515   2516#
                    2522   2527#  2532   2533#  2539   2544#  2549   2550#  2557   2562#  2567   2568#  2575
                    2580#  2585   2586#  2593   2598#  2603   2604#  2611   2616#  2621   2622#  2629   2634#
                    2639   2640#  2646   2651#  2656   2657#  2663   2668#  2673   2678#  2683   2684#  2690
                    2695#  2700   2701#  2707   2712#  2717   2722#  2727   2732#  2737   2738#  2744   2749#
                    2754   2759#  2764   2769#  2774   2775#  2781   2786#  2791   2796#  2801   2802#  2808
                    2813#  2818   2819#  2825   2830#  2835   2836#  2843   2848#  2853   2854#  2860   2865#
                    2870   2871#  2878   2883#  2888   2889#  2895   2900#  2905   2910#  2915   2920#  2925
                    2926#  2932   2937#  2942   2943#  2949   2954#  2959   2960#  2966   2971#  2976   2977#
                    2983   2990#  2995   2997#  3002
T$CODE= 003032      8410#  8416#  8421#  8426#
T$ERRN= 000027      1368#  2185#  3293#  3322#  3367#  3390#  3431#  3443#  3486#  3503#  3539#  3556#  3589#
                    3606#  3641#  3653#  3698#  3721#  3762#  3786#  3827#  3850#  3892#  3910#  3952#  3996#
                    4026#  4074#  4104#  4152#  4182#  4230#  4260#  4312#  4347#  4400#  4435#  4484#  4514#
                    4562#  4592#  4640#  4670#  4718#  4748#  4796#  4826#  4874#  4904#  4952#  4982#  5030#
                    5060#  5116#  5144#  5193#  5224#  5284#  5321#  5384#  5416#  5464#  5470#  5509#  5515#
                    5569#  5619#  5627#  5675#  5727#  5777#  5848#  5902#  5951#  5966#  6013#  6061#  6132#
```

T$EXCP= 000000
T$FLAG= 000040
T$GMAN= 000000
T$HILI= 000007
T$LAST= 000001
T$LOLI= 000004
T$LSYM= 010000
T$LTNO= 000114
T$NEST= 177777

6194# 6263# 6332# 6401# 6470# 6539# 6608# 6677# 6746# 6817# 6887# 6958# 7028#
7098# 7168# 7239# 7309# 7380# 7450# 7520# 7590# 7660# 7730# 7800# 7870# 7940#
8011# 8081# 8151# 8221# 8290# 8360#
8410# 8415 8416# 8420 8421# 8425 8426# 8431
3022# 3024 3283# 3300# 3371# 3394# 3435# 3447# 3490# 3507# 3543# 3593# 3645#
3657# 3702# 3725# 3766# 3790# 3831# 3854# 3896# 3914# 3957# 4000# 4030# 4078#
4108# 4156# 4186# 4234# 4264# 4316# 4351# 4404# 4439# 4488# 4518# 4566# 4596#
4644# 4674# 4722# 4752# 4800# 4830# 4878# 4908# 4956# 4986# 5034# 5064# 5120#
5148# 5197# 5228# 5288# 5325# 5388# 5420# 5679# 5682# 5731# 5734# 5781# 5784#
6065# 6081# 6136# 6198# 6208# 6267# 6277# 6336# 6346# 6405# 6415# 6474# 6484#
6543# 6553# 6612# 6622# 6681# 6691# 6750# 6760# 6821# 6831# 6891# 6901# 6962#
6972# 7032# 7042# 7102# 7112# 7172# 7182# 7243# 7253# 7313# 7323# 7384# 7394#
7454# 7464# 7524# 7534# 7594# 7604# 7664# 7674# 7734# 7744# 7804# 7814# 7874#
7884# 7944# 7954# 8015# 8025# 8085# 8095# 8155# 8165# 8225# 8235# 8294# 8304#
8364# 8374#
1368#
8410# 8414 8416# 8419 8421# 8424 8426# 8430
1368# 8542#
8410# 8413 8416# 8418 8421# 8423 8426# 8429
1368# 1617 1637 2507 2524 2541 2559 2577 2595 2613 2631 2648 2665
2675 2692 2709 2719 2729 2746 2756 2766 2783 2793 2810 2827 2845
2862 2880 2897 2907 2917 2934 2951 2968 2985 3005 3028 3175 3201
3217 3237 3255 3304 3328 3405 3456 3515 3565 3615 3666 3736 3801
3865 3921 3962 4041 4119 4197 4275 4363 4451 4529 4607 4685 4763
4841 4919 4997 5075 5160 5241 5340 5343 5431 5477 5522 5577 5636
5688 5740 5790 5854 5911 5973 6019 6088 6143 6217 6286 6355 6424
6493 6562 6631 6700 6770 6840 6911 6981 7051 7121 7191 7262 7332
7403 7473 7543 7613 7683 7753 7823 7893 7963 8034 8104 8174 8244
8313 8383 8439 8525
8545#
1368# 1374# 1490# 1495# 1598# 1616# 1630# 1636# 2492# 2506# 2509# 2523# 2526#
2540# 2543# 2558# 2561# 2576# 2579# 2594# 2597# 2612# 2615# 2630# 2633# 2647#
2650# 2664# 2667# 2674# 2677# 2691# 2694# 2708# 2711# 2718# 2721# 2728# 2731#
2745# 2748# 2755# 2758# 2765# 2768# 2782# 2785# 2792# 2795# 2809# 2812# 2826#
2829# 2844# 2847# 2861# 2864# 2879# 2882# 2896# 2899# 2906# 2909# 2916# 2919#
2933# 2936# 2950# 2953# 2967# 2970# 2984# 2988# 3004# 3018# 3027# 3042# 3174#
3179# 3200# 3211# 3216# 3231# 3236# 3252# 3254# 3276# 3303# 3314# 3327# 3352#
3359# 3378# 3383# 3401# 3404# 3417# 3422# 3452# 3455# 3474# 3477# 3493# 3496#
3510# 3514# 3527# 3530# 3546# 3549# 3560# 3564# 3577# 3580# 3596# 3599# 3610#
3614# 3626# 3631# 3662# 3665# 3684# 3690# 3708# 3712# 3732# 3735# 3748# 3754#
3772# 3776# 3797# 3800# 3813# 3819# 3840# 3843# 3861# 3864# 3877# 3883# 3899#
3903# 3917# 3920# 3937# 3961# 3974# 3982# 4006# 4011# 4037# 4040# 4053# 4060#
4084# 4089# 4115# 4118# 4131# 4138# 4162# 4167# 4193# 4196# 4209# 4216# 4240#
4245# 4271# 4274# 4287# 4294# 4323# 4328# 4359# 4362# 4375# 4382# 4411# 4416#
4447# 4450# 4463# 4470# 4494# 4499# 4525# 4528# 4541# 4548# 4572# 4577# 4603#
4606# 4619# 4626# 4650# 4655# 4681# 4684# 4697# 4704# 4728# 4733# 4759# 4762#
4775# 4782# 4806# 4811# 4837# 4840# 4853# 4860# 4884# 4889# 4915# 4918# 4931#
4938# 4962# 4967# 4993# 4996# 5009# 5016# 5040# 5045# 5071# 5074# 5087# 5094#
5127# 5132# 5156# 5159# 5172# 5179# 5203# 5208# 5237# 5240# 5253# 5261# 5263#
5294# 5298# 5332# 5339# 5342# 5355# 5362# 5395# 5400# 5427# 5430# 5442# 5476#
5488# 5521# 5534# 5576# 5589# 5635# 5647# 5687# 5699# 5739# 5751# 5789# 5802#
5853# 5870# 5910# 5923# 5972# 5985# 6018# 6031# 6087# 6103# 6112# 6139# 6142#
6162# 6173# 6205# 6216# 6231# 6242# 6274# 6285# 6300# 6311# 6343# 6354# 6369#
6380# 6412# 6423# 6438# 6449# 6481# 6492# 6507# 6518# 6550# 6561# 6576# 6587#
6619# 6630# 6645# 6656# 6688# 6699# 6714# 6725# 6757# 6769# 6785# 6796# 6828#
6839# 6855# 6866# 6898# 6910# 6926# 6937# 6969# 6980# 6996# 7007# 7039# 7050#

```
                    7066#  7077#  7109#  7120#  7136#  7147#  7179#  7190#  7207#  7218#  7250#  7261#  7277#
                    7288#  7320#  7331#  7348#  7359#  7391#  7402#  7418#  7429#  7461#  7472#  7488#  7499#
                    7531#  7542#  7558#  7569#  7601#  7612#  7628#  7639#  7671#  7682#  7698#  7709#  7741#
                    7752#  7768#  7779#  7811#  7822#  7838#  7849#  7881#  7892#  7908#  7919#  7951#  7962#
                    7979#  7990#  8022#  8033#  8049#  8060#  8092#  8103#  8119#  8130#  8162#  8173#  8189#
                    8200#  8232#  8243#  8258#  8269#  8301#  8312#  8328#  8339#  8371#  8382#  8406#  8437#
                    8518#  8523#  8533#
T$NS0 = 000000      1374#  8533
T$NS1 = 000005      1490#  1495   1598#  1616   1630#  1636   2492#  2506   2509#  2523   2526#  2540   2543#
                    2558   2561#  2576   2579#  2594   2597#  2612   2615#  2630   2633#  2647   2650#  2664
                    2667#  2674   2677#  2691   2694#  2708   2711#  2718   2721#  2728   2731#  2745   2748#
                    2755   2758#  2765   2768#  2782   2785#  2792   2795#  2809   2812#  2826   2829#  2844
                    2847#  2861   2864#  2879   2882#  2896   2899#  2906   2909#  2916   2919#  2933   2936#
                    2950   2953#  2967   2970#  2984   2988#  3004   3018#  3027   3042#  3174   3179#  3200
                    3211#  3216   3231#  3236   3252#  3254   3276#  3303   3314#  3327   3352#  3404   3417#
                    3455   3474#  3514   3527#  3564   3577#  3614   3626#  3665   3684#  3735   3748#  3800
                    3813#  3864   3877#  3920   3937#  3961   3974#  4040   4053#  4118   4131#  4196   4209#
                    4274   4287#  4362   4375#  4450   4463#  4528   4541#  4606   4619#  4684   4697#  4762
                    4775#  4840   4853#  4918   4931#  4996   5009#  5074   5087#  5159   5172#  5240   5253#
                    5342   5355#  5430   5442#  5476   5488#  5521   5534#  5576   5589#  5635   5647#  5687
                    5699#  5739   5751#  5789   5802#  5853   5870#  5910   5923#  5972   5985#  6018   6031#
                    6087   6103#  6142   6162#  6216   6231#  6285   6300#  6354   6369#  6423   6438#  6492
                    6507#  6561   6576#  6630   6645#  6699   6714#  6769   6785#  6839   6855#  6910   6926#
                    6980   6996#  7050   7066#  7120   7136#  7190   7207#  7261   7277#  7331   7348#  7402
                    7418#  7472   7488#  7542   7558#  7612   7628#  7682   7698#  7752   7768#  7822   7838#
                    7892   7908#  7962   7979#  8033   8049#  8103   8119#  8173   8189#  8243   8258#  8312
                    8328#  8382   8406#  8437   8518#  8523
T$NS2 = 000003       3359#  3378   3383#  3401   3422#  3452   3477#  3493   3496#  3510   3530#  3546   3549#
                     3560   3580#  3596   3599#  3610   3631#  3662   3690#  3708   3712#  3732   3754#  3772
                     3776#  3797   3819#  3840   3843#  3861   3883#  3899   3903#  3917   3982#  4006   4011#
                     4037   4060#  4084   4089#  4115   4138#  4162   4167#  4193   4216#  4240   4245#  4271
                     4294#  4323   4328#  4359   4382#  4411   4416#  4447   4470#  4494   4499#  4525   4548#
                     4572   4577#  4603   4626#  4650   4655#  4681   4704#  4728   4733#  4759   4782#  4806
                     4811#  4837   4860#  4884   4889#  4915   4938#  4962   4967#  4993   5016#  5040   5045#
                     5071   5094#  5127   5132#  5156   5179#  5203   5208#  5237   5261#  5339   5362#  5395
                     5400#  5427   6112#  6139   6173#  6205   6242#  6274   6311#  6343   6380#  6412   6449#
                     6481   6518#  6550   6587#  6619   6656#  6688   6725#  6757   6796#  6828   6866#  6898
                     6937#  6969   7007#  7039   7077#  7109   7147#  7179   7218#  7250   7288#  7320   7359#
                     7391   7429#  7461   7499#  7531   7569#  7601   7639#  7671   7709#  7741   7779#  7811
                     7849#  7881   7919#  7951   7990#  8022   8060#  8092   8130#  8162   8200#  8232   8269#
                     8301   8339#  8371
T$NS3 = 000003       5263#  5294   5298#  5332
T$PTNU= 000000       1368#
T$SAVL= 177777       1368#
T$SEGL= 177777       1368#  3359#  3372   3378#  3380   3383#  3395   3401#  3403   3422#  3436   3448   3452#
                     3454   3477#  3491   3493#  3495   3496#  3508   3510#  3512   3530#  3544   3546#  3548
                     3549#  3560#  3562   3580#  3594   3596#  3598   3599#  3610#  3612   3631#  3646   3658
                     3662#  3664   3690#  3703   3708#  3710   3712#  3726   3732#  3734   3754#  3767   3772#
                     3774   3776#  3791   3797#  3799   3819#  3832   3840#  3842   3843#  3855   3861#  3863
                     3883#  3897   3899#  3901   3903#  3915   3917#  3919   3982#  4001   4006#  4008   4011#
                     4031   4037#  4039   4060#  4079   4084#  4086   4089#  4109   4115#  4117   4138#  4157
                     4162#  4164   4167#  4187   4193#  4195   4216#  4235   4240#  4242   4245#  4265   4271#
                     4273   4294#  4317   4323#  4325   4328#  4352   4359#  4361   4382#  4405   4411#  4413
                     4416#  4440   4447#  4449   4470#  4489   4494#  4496   4499#  4519   4525#  4527   4548#
                     4567   4572#  4574   4577#  4597   4603#  4605   4626#  4645   4650#  4652   4655#  4675
                     4681#  4683   4704#  4723   4728#  4730   4733#  4753   4759#  4761   4782#  4801   4806#
```

```
                    4808   4811#  4831   4837#  4839   4860#  4879   4884#  4886   4889#  4909   4915#  4917
                    4938#  4957   4962#  4964   4967#  4987   4993#  4995   5016#  5035   5040#  5042   5045#
                    5065   5071#  5073   5094#  5121   5127#  5129   5132#  5149   5156#  5158   5179#  5198
                    5203#  5205   5208#  5229   5237#  5239   5263#  5289   5294#  5296   5298#  5332#  5334
                    5362#  5389   5395#  5397   5400#  5421   5427#  5429   6112#  6137   6139#  6141   6173#
                    6199   6205#  6207   6242#  6268   6274#  6276   6311#  6337   6343#  6345   6380#  6406
                    6412#  6414   6449#  6475   6481#  6483   6518#  6544   6550#  6552   6587#  6613   6619#
                    6621   6656#  6682   6688#  6690   6725#  6751   6757#  6759   6796#  6822   6828#  6830
                    6866#  6892   6898#  6900   6937#  6963   6969#  6971   7007#  7033   7039#  7041   7077#
                    7103   7109#  7111   7147#  7173   7179#  7181   7218#  7244   7250#  7252   7288#  7314
                    7320#  7322   7359#  7385   7391#  7393   7429#  7455   7461#  7463   7499#  7525   7531#
                    7533   7569#  7595   7601#  7603   7639#  7665   7671#  7673   7709#  7735   7741#  7743
                    7779#  7805   7811#  7813   7849#  7875   7881#  7883   7919#  7945   7951#  7953   7990#
                    8016   8022#  8024   8060#  8086   8092#  8094   8130#  8156   8162#  8164   8200#  8226
                    8232#  8234   8269#  8295   8301#  8303   8339#  8365   8371#  8373
T$SEK0= 010000      3359#  3372   3378   3383#  3395   3401   3422#  3436   3448   3452   3474#  3491   3493
                    3496#  3508   3510   3530#  3544   3546   3549#  3560   3580#  3594   3596   3599#  3610
                    3631#  3646   3658   3662   3690#  3703   3708   3712#  3726   3732   3754#  3767   3772
                    3776#  3791   3797   3819#  3832   3840   3843#  3855   3861   3883#  3897   3899   3903#
                    3915   3917   3982#  4001   4006   4011#  4031   4037   4060#  4079   4084   4089#  4109
                    4115   4138#  4157   4162   4167#  4187   4193   4216#  4235   4240   4245#  4265   4271
                    4294#  4317   4323   4328#  4352   4359   4382#  4405   4411   4416#  4440   4447   4470#
                    4489   4494   4499#  4519   4525   4548#  4567   4572   4577#  4597   4603   4626#  4645
                    4650   4655#  4675   4681   4704#  4723   4728   4733#  4753   4759   4782#  4801   4806
                    4811#  4831   4837   4860#  4879   4884   4889#  4909   4915   4938#  4957   4962   4967#
                    4987   4993   5016#  5035   5040   5045#  5065   5071   5094#  5121   5127   5132#  5149
                    5156   5179#  5198   5203   5208#  5229   5237   5263#  5289   5294   5298#  5332   5362#
                    5389   5395   5400#  5421   5427   6112#  6137   6139   6173#  6199   6205   6242#  6268
                    6274   6311#  6337   6343   6380#  6406   6412   6449#  6475   6481   6518#  6544   6550
                    6587#  6613   6619   6656#  6682   6688   6725#  6751   6757   6796#  6822   6828   6866#
                    6892   6898   6937#  6963   6969   7007#  7033   7039   7077#  7103   7109   7147#  7173
                    7179   7218#  7244   7250   7288#  7314   7320   7359#  7385   7391   7429#  7455   7461
                    7499#  7525   7531   7569#  7595   7601   7639#  7665   7671   7709#  7735   7741   7779#
                    7805   7811   7849#  7875   7881   7919#  7945   7951   7990#  8016   8022   8060#  8086
                    8092   8130#  8156   8162   8200#  8226   8232   8269#  8295   8301   8339#  8365   8371
T$SUBN= 000000      1368#  3275#  3313#  3351#  3416#  3473#  3526#  3576#  3625#  3683#  3747#  3812#  3876#
                    3936#  3973#  4052#  4130#  4208#  4286#  4374#  4462#  4540#  4618#  4696#  4774#  4852#
                    4930#  5008#  5086#  5171#  5252#  5260#  5354#  5441#  5487#  5533#  5588#  5646#  5698#
                    5750#  5801#  5869#  5922#  5984#  6030#  6102#  6161#  6230#  6299#  6368#  6437#  6506#
                    6575#  6644#  6713#  6784#  6854#  6925#  6995#  7065#  7135#  7206#  7276#  7347#  7417#
                    7487#  7557#  7627#  7697#  7767#  7837#  7907#  7978#  8048#  8118#  8188#  8257#  8327#
T$TAGL= 177777      1368#
T$TAGN= 010171      1368#  1490#  1598#  1630#  2492#  2509#  2526#  2543#  2561#  2579#  2597#  2615#  2633#
                    2650#  2667#  2677#  2694#  2711#  2721#  2731#  2748#  2758#  2768#  2785#  2795#  2812#
                    2829#  2847#  2864#  2882#  2899#  2909#  2919#  2936#  2953#  2970#  2988#  3018#  3042#
                    3179#  3211#  3231#  3252#  3276#  3314#  3352#  3417#  3474#  3527#  3577#  3626#  3684#
                    3748#  3813#  3877#  3937#  3974#  4053#  4131#  4209#  4287#  4375#  4463#  4541#  4619#
                    4697#  4775#  4853#  4931#  5009#  5087#  5172#  5253#  5261#  5355#  5442#  5488#  5534#
                    5589#  5647#  5699#  5751#  5802#  5870#  5923#  5985#  6031#  6103#  6162#  6231#  6300#
                    6369#  6438#  6507#  6576#  6645#  6714#  6785#  6855#  6926#  6996#  7066#  7136#  7207#
                    7277#  7348#  7418#  7488#  7558#  7628#  7698#  7768#  7838#  7908#  7979#  8049#  8119#
                    8189#  8258#  8328#  8406#  8518#
T$TEMP= 000000      1495#  1506#  1507#  1508#  1509#  1510#  1511#  1512#  1513#  1514#  1515#  1516#  1517#
                    1518#  1519#  1520#  1521#  1522#  1523#  1524#  1525#  1526#  1527#  1528#  1529#  1530#
                    1531#  1532#  1533#  1534#  1535#  1536#  1537#  1538#  1539#  1540#  1541#  1542#  1543#
                    1544#  1545#  1546#  1547#  1548#  1549#  1550#  1551#  1552#  1553#  1554#  1555#  1556#
```

	1557#	1558#	1559#	1560#	1561#	1562#	1563#	1564#	1565#	1566#	1567#	1568#	1569#
	1570#	1571#	1572#	1573#	1574#	1575#	1576#	1577#	1578#	1579#	1580#	1581#	1582#
	1616#	1636#	2506#	2523#	2540#	2558#	2576#	2594#	2612#	2630#	2647#	2664#	2674#
	2691#	2708#	2718#	2728#	2745#	2755#	2765#	2782#	2792#	2809#	2826#	2844#	2861#
	2879#	2896#	2906#	2916#	2933#	2950#	2967#	2984#	3004#	3022#	3023	3027#	3174#
	3200#	3216#	3236#	3254#	3283#	3284	3300#	3301	3303#	3327#	3371#	3372#	3378#
	3394#	3395#	3401#	3404#	3435#	3436#	3447#	3448#	3452#	3455#	3490#	3491#	3493#
	3507#	3508#	3510#	3514#	3543#	3544#	3546#	3560#	3564#	3593#	3594#	3596#	3610#
	3614#	3645#	3646#	3657#	3658#	3662#	3665#	3702#	3703#	3708#	3725#	3726#	3732#
	3735#	3766#	3767#	3772#	3790#	3791#	3797#	3800#	3831#	3832#	3840#	3854#	3855#
	3861#	3864#	3896#	3897#	3899#	3914#	3915#	3917#	3920#	3957#	3958	3961#	4000#
	4001#	4006#	4030#	4031#	4037#	4040#	4078#	4079#	4084#	4108#	4109#	4115#	4118#
	4156#	4157#	4162#	4186#	4187#	4193#	4196#	4234#	4235#	4240#	4264#	4265#	4271#
	4274#	4316#	4317#	4323#	4351#	4352#	4359#	4362#	4404#	4405#	4411#	4439#	4440#
	4447#	4450#	4488#	4489#	4494#	4518#	4519#	4525#	4528#	4566#	4567#	4572#	4596#
	4597#	4603#	4606#	4644#	4645#	4650#	4674#	4675#	4681#	4684#	4722#	4723#	4728#
	4752#	4753#	4759#	4762#	4800#	4801#	4806#	4830#	4831#	4837#	4840#	4878#	4879#
	4884#	4908#	4909#	4915#	4918#	4956#	4957#	4962#	4986#	4987#	4993#	4996#	5034#
	5035#	5040#	5064#	5065#	5071#	5074#	5120#	5121#	5127#	5148#	5149#	5156#	5159#
	5197#	5198#	5203#	5228#	5229#	5237#	5240#	5288#	5289#	5294#	5325#	5326	5332#
	5339#	5342#	5388#	5389#	5395#	5420#	5421#	5427#	5430#	5476#	5521#	5576#	5635#
	5679#	5680	5682#	5683	5687#	5731#	5732	5734#	5735	5739#	5781#	5782	5784#
	5785	5789#	5853#	5910#	5972#	6018#	6065#	6066	6081#	6082	6087#	6136#	6137#
	6139#	6142#	6198#	6199#	6205#	6208#	6209	6216#	6267#	6268#	6274#	6277#	6278
	6285#	6336#	6337#	6343#	6346#	6347	6354#	6405#	6406#	6412#	6415#	6416	6423#
	6474#	6475#	6481#	6484#	6485	6492#	6543#	6544#	6550#	6553#	6554	6561#	6612#
	6613#	6619#	6622#	6623	6630#	6681#	6682#	6688#	6691#	6692	6699#	6750#	6751#
	6757#	6760#	6761	6769#	6821#	6822#	6828#	6831#	6832	6839#	6891#	6892#	6898#
	6901#	6902	6910#	6962#	6963#	6969#	6972#	6973	6980#	7032#	7033#	7039#	7042#
	7043	7050#	7102#	7103#	7109#	7112#	7113	7120#	7172#	7173#	7179#	7182#	7183
	7190#	7243#	7244#	7250#	7253#	7254	7261#	7313#	7314#	7320#	7323#	7324	7331#
	7384#	7385#	7391#	7394#	7395	7402#	7454#	7455#	7461#	7464#	7465	7472#	7524#
	7525#	7531#	7534#	7535	7542#	7594#	7595#	7601#	7604#	7605	7612#	7664#	7665#
	7671#	7674#	7675	7682#	7734#	7735#	7741#	7744#	7745	7752#	7804#	7805#	7811#
	7814#	7815	7822#	7874#	7875#	7881#	7884#	7885	7892#	7944#	7945#	7951#	7954#
	7955	7962#	8015#	8016#	8022#	8025#	8026	8033#	8085#	8086#	8092#	8095#	8096
	8103#	8155#	8156#	8162#	8165#	8166	8173#	8225#	8226#	8232#	8235#	8236	8243#
	8294#	8295#	8301#	8304#	8305	8312#	8364#	8365#	8371#	8374#	8375	8382#	8410#
	8416#	8421#	8426#	8437#	8523#	8533#							
T$TEST= 000114	1368#	3268	3272	3275#	3307	3310	3313#	3331	3348	3351#	3408	3413	3416#
	3465	3470	3473#	3518	3523	3526#	3568	3573	3576#	3618	3622	3625#	3674
	3679	3683#	3739	3744	3747#	3804	3809	3812#	3868	3873	3876#	3926	3933
	3936#	3965	3970	3973#	4044	4049	4052#	4122	4127	4130#	4200	4205	4208#
	4278	4283	4286#	4366	4371	4374#	4454	4459	4462#	4532	4537	4540#	4610
	4615	4618#	4688	4693	4696#	4766	4771	4774#	4844	4849	4852#	4922	4927
	4930#	5000	5005	5008#	5078	5083	5086#	5163	5168	5171#	5244	5249	5252#
	5260	5346	5351	5354#	5434	5438	5441#	5480	5484	5487#	5525	5530	5533#
	5580	5585	5588#	5639	5643	5646#	5691	5695	5698#	5743	5747	5750#	5793
	5798	5801#	5857	5866	5869#	5914	5919	5922#	5976	5981	5984#	6022	6027
	6030#	6095	6099	6102#	6151	6158	6161#	6220	6227	6230#	6289	6296	6299#
	6358	6365	6368#	6427	6434	6437#	6496	6503	6506#	6565	6572	6575#	6634
	6641	6644#	6703	6710	6713#	6774	6781	6784#	6844	6851	6854#	6915	6922
	6925#	6985	6992	6995#	7055	7062	7065#	7125	7132	7135#	7195	7203	7206#
	7266	7273	7276#	7336	7343	7347#	7407	7414	7417#	7477	7484	7487#	7547
	7554	7557#	7617	7625	7627#	7687	7694	7697#	7757	7764	7767#	7827	7834
	7837#	7897	7904	7907#	7967	7974	7978#	8038	8045	8048#	8108	8115	8118#

```
                    8178  8185  8188#  8247  8254  8257#  8317  8324  8327#  8545
T$TSTM= 177777      1368#  2184  2497  2504  2507  2514  2521  2524  2531  2538  2541  2548  2556
                    2559  2566  2574  2577  2584  2592  2595  2602  2610  2613  2620  2628  2631
                    2638  2645  2648  2655  2662  2665  2672  2675  2682  2689  2692  2699  2706
                    2709  2716  2719  2726  2729  2736  2743  2746  2753  2756  2763  2766  2773
                    2780  2783  2790  2793  2800  2807  2810  2817  2824  2827  2834  2842  2845
                    2852  2859  2862  2869  2877  2880  2887  2894  2897  2904  2907  2914  2917
                    2924  2931  2934  2941  2948  2951  2958  2965  2968  2975  2982  2985  2994
                    3001  3005  3028  3058  3064  3070  3077  3093  3175  3195  3201  3213  3217
                    3234  3237  3255  3283  3292  3300  3304  3321  3328  3359  3366  3371  3379
                    3383  3389  3394  3402  3405  3422  3430  3435  3442  3447  3453  3456  3477
                    3485  3490  3494  3496  3502  3507  3511  3515  3530  3538  3543  3547  3549
                    3555  3561  3565  3580  3588  3593  3597  3599  3605  3611  3615  3631  3640
                    3645  3652  3657  3663  3666  3690  3697  3702  3709  3712  3720  3725  3733
                    3736  3754  3761  3766  3773  3776  3785  3790  3798  3801  3819  3826  3831
                    3841  3843  3849  3854  3862  3865  3883  3891  3896  3900  3903  3909  3914
                    3918  3921  3951  3957  3962  3982  3995  4000  4007  4011  4025  4030  4038
                    4041  4060  4073  4078  4085  4089  4103  4108  4116  4119  4138  4151  4156
                    4163  4167  4181  4186  4194  4197  4216  4229  4234  4241  4245  4259  4264
                    4272  4275  4294  4311  4316  4324  4328  4346  4351  4360  4363  4382  4399
                    4404  4412  4416  4434  4439  4448  4451  4470  4483  4488  4495  4499  4513
                    4518  4526  4529  4548  4561  4566  4573  4577  4591  4596  4604  4607  4626
                    4639  4644  4651  4655  4669  4674  4682  4685  4704  4717  4722  4729  4733
                    4747  4752  4760  4763  4782  4795  4800  4807  4811  4825  4830  4838  4841
                    4860  4873  4878  4885  4889  4903  4908  4916  4919  4938  4951  4956  4963
                    4967  4981  4986  4994  4997  5016  5029  5034  5041  5045  5059  5064  5072
                    5075  5094  5115  5120  5128  5132  5143  5148  5157  5160  5179  5192  5197
                    5204  5208  5223  5228  5238  5241  5261  5263  5283  5288  5295  5298  5320
                    5325  5333  5340  5343  5362  5383  5388  5396  5400  5415  5420  5428  5431
                    5445  5453  5460  5463  5469  5477  5498  5505  5508  5514  5522  5541  5559
                    5566  5568  5577  5596  5615  5618  5626  5636  5648  5674  5679  5682  5688
                    5726  5731  5734  5740  5776  5781  5784  5790  5847  5854  5901  5911  5950
                    5965  5973  6012  6019  6060  6065  6081  6088  6112  6131  6136  6140  6143
                    6173  6193  6198  6206  6208  6217  6242  6262  6267  6275  6277  6286  6311
                    6331  6336  6344  6346  6355  6380  6400  6405  6413  6415  6424  6449  6469
                    6474  6482  6484  6493  6518  6538  6543  6551  6553  6562  6587  6607  6612
                    6620  6622  6631  6656  6676  6681  6689  6691  6700  6725  6745  6750  6758
                    6760  6770  6796  6816  6821  6829  6831  6840  6866  6886  6891  6899  6901
                    6911  6937  6957  6962  6970  6972  6981  7007  7027  7032  7040  7042  7051
                    7077  7097  7102  7110  7112  7121  7147  7167  7172  7180  7182  7191  7218
                    7238  7243  7251  7253  7262  7288  7308  7313  7321  7323  7332  7359  7379
                    7384  7392  7394  7403  7429  7449  7454  7462  7464  7473  7499  7519  7524
                    7532  7534  7543  7569  7589  7594  7602  7604  7613  7639  7659  7664  7672
                    7674  7683  7709  7729  7734  7742  7744  7753  7779  7799  7804  7812  7814
                    7823  7849  7869  7874  7882  7884  7893  7919  7939  7944  7952  7954  7963
                    7990  8010  8015  8023  8025  8034  8060  8080  8085  8093  8095  8104  8130
                    8150  8155  8163  8165  8174  8200  8220  8225  8233  8235  8244  8269  8289
                    8294  8302  8304  8313  8339  8359  8364  8372  8374  8383
T$TSTS= 000001       1368#  3276#  3314#  3352#  3417#  3474#  3527#  3577#  3626#  3684#  3748#  3813#  3877#
                    3937#  3974#  4053#  4131#  4209#  4287#  4375#  4463#  4541#  4619#  4697#  4775#  4853#
                    4931#  5009#  5087#  5172#  5253#  5355#  5442#  5488#  5534#  5589#  5647#  5699#  5751#
                    5802#  5870#  5923#  5985#  6031#  6103#  6162#  6231#  6300#  6369#  6438#  6507#  6576#
                    6645#  6714#  6785#  6855#  6926#  6996#  7066#  7136#  7207#  7277#  7348#  7418#  7488#
                    7558#  7628#  7698#  7768#  7838#  7908#  7979#  8049#  8119#  8189#  8258#  8328#
T$$AU = 010051       3252#  3254
T$$AUT= 010046       3179#  3200
```

```
T$$CLE= 010047          3211#  3216
T$$DU = 010050          3231#  3236
T$$HAR= 010167          8406#  8438
T$$HW = 010001          1598#  1616
T$$INI= 010045          3042#  3174
T$$MSG= 010043          2492#  2506  2509#  2523  2526#  2540  2543#  2558  2561#  2576  2579#  2594  2597#
                        2612   2615#  2630   2633#  2647   2650#  2664   2667#  2674   2677#  2691   2694#  2708
                        2711#  2718   2721#  2728   2731#  2745   2748#  2755   2758#  2765   2768#  2782   2785#
                        2792   2795#  2809   2812#  2826   2829#  2844   2847#  2861   2864#  2879   2882#  2896
                        2899#  2906   2909#  2916   2919#  2933   2936#  2950   2953#  2967   2970#  2984   2988#
                        3004
T$$PRO= 010000          1490#
T$$RPT= 010044          3018#  3022  3027
T$$SEG= 010000          3359#  3371  3378#  3383#  3394   3401#  3422#  3435   3447   3452#  3477#  3490   3493#
                        3496#  3507  3510#  3530#  3543   3546#  3549#  3560#  3580#  3593   3596#  3599#  3610#
                        3631#  3645  3657   3662#  3690#  3702   3708#  3712#  3725   3732#  3754#  3766   3772#
                        3776#  3790  3797#  3819#  3831   3840#  3843#  3854   3861#  3883#  3896   3899#  3903#
                        3914   3917# 3982#  4000   4006#  4011#  4030   4037#  4060#  4078   4084#  4089#  4108
                        4115#  4133# 4156   4162#  4167#  4186   4193#  4216#  4234   4240#  4245#  4264   4271#
                        4294#  4316  4323#  4328#  4351   4359#  4382#  4404   4411#  4416#  4439   4447#  4470#
                        4488   4494# 4499#  4518   4525#  4548#  4566   4572#  4577#  4596   4603#  4626#  4644
                        4650#  4655# 4674   4681#  4704#  4722   4728#  4733#  4752   4759#  4782#  4800   4806#
                        4811#  4830  4837#  4860#  4878   4884#  4889#  4908   4915#  4938#  4956   4962#  4967#
                        4986   4993# 5016#  5034   5040#  5045#  5064   5071#  5094#  5120   5127#  5132#  5148
                        5156#  5179# 5197   5203#  5208#  5228   5237#  5263#  5288   5294#  5298#  5332#  5362#
                        5388   5395# 5400#  5420   5427#  6112#  6136   6139#  6173#  6198   6205#  6242#  6267
                        6274#  6311# 6336   6343#  6380#  6405   6412#  6449#  6474   6481#  6518#  6543   6550#
                        6587#  6612  6619#  6656#  6681   6688#  6725#  6750   6757#  6796#  6821   6828#  6866#
                        6891   6898# 6937#  6962   6969#  7007#  7032   7039#  7077#  7102   7109#  7147#  7172
                        7179#  7218# 7243   7250#  7288#  7313   7320#  7359#  7384   7391#  7429#  7454   7461#
                        7499#  7524  7531#  7569#  7594   7601#  7639#  7664   7671#  7709#  7734   7741#  7779#
                        7804   7811# 7849#  7874   7881#  7919#  7944   7951#  7990#  8015   8022#  8060#  8085
                        8092#  8130# 8155   8162#  8200#  8225   8232#  8269#  8294   8301#  8339#  8364   8371#
T$$SOF= 010170          8518#  8524
T$$SUB= 010110          5261#  5339
T$$SW = 010002          1630#  1636
T$$TES=  10166          3276#  3283  3300   3303   3314#  3327   3352#  3404   3417#  3455   3474#  3514   3527#
                        3564   3577# 3614   3626#  3665   3684#  3735   3748#  3800   3813#  3864   3877#  3920
                        3937#  3957  3961   3974#  4040   4053#  4118   4131#  4196   4209#  4274   4287#  4362
                        4375#  4450  4463#  4528   4541#  4606   4619#  4684   4697#  4762   4775#  4840   4853#
                        4918   4931# 4996   5009#  5074   5087#  5159   5172#  5240   5253#  5325   5342   5355#
                        5430   5442# 5476   5488#  5521   5534#  5576   5589#  5635   5647#  5679   5682   5687
                        5699#  5731  5734   5739   5751#  5781   5784   5789   5802#  5853   5870#  5910   5923#
                        5972   5985# 6018   6031#  6065   6081   6087   6103#  6142   6162#  6208   6216   6231#
                        6277   6285  6300#  6346   6354   6369#  6415   6423   6438#  6484   6492   6507#  6553
                        6561   6576# 6622   6630   6645#  6691   6699   6714#  6760   6769   6785#  6831   6839
                        6855#  6901  6910   6926#  6972   6980   6996#  7042   7050   7066#  7112   7120   7136#
                        7182   7190  7207#  7253   7261   7277#  7323   7331   7348#  7394   7402   7418#  7464
                        7472   7488# 7534   7542   7558#  7604   7612   7628#  7674   7682   7698#  7744   7752
                        7768#  7814  7822   7838#  7884   7892   7908#  7954   7962   7979#  8025   8033   8049#
                        8095   8103  8119#  8165   8173   8189#  8235   8243   8258#  8304   8312   8328#  8374
                        8382
T1      011366 G        1506   3275#
T10     013054 G        1515   3747#
T11     013240 G        1516   3812#
T12     013426 G        1517   3876#
```

T13	013576	G	1518	3936#
T14	013704	G	1519	3973#
T15	014134	G	1520	4052#
T16	014364	G	1521	4130#
T17	014614	G	1522	4208#
T18	015044	G	1523	4286#
T19	015340	G	1524	4374#
T2	011514	G	1507	3313#
T20	015634	G	1525	4462#
T21	016064	G	1526	4540#
T22	016314	G	1527	4618#
T23	016544	G	1528	4696#
T24	016774	G	1529	4774#
T25	017224	G	1530	4852#
T26	017454	G	1531	4930#
T27	017704	G	1532	5008#
T28	020134	G	1533	5086#
T29	020432	G	1534	5171#
T3	011560	G	1508	3351#
T30	020662	G	1535	5252#
T30.1	020700		5260#	
T31	021226	G	1536	5354#
T32	021540	G	1537	5441#
T33	021702	G	1538	5487#
T34	022044	G	1539	5533#
T35	022220	G	1540	5588#
T36	022424	G	1541	5646#
T37	022570	G	1542	5698#
T38	022740	G	1543	5750#
T39	023104	G	1544	5801#
T4	011746	G	1509	3416#
T40	023272	G	1545	5869#
T41	023506	G	1546	5922#
T42	023724	G	1547	5984#
T43	024056	G	1548	6030#
T44	024274	G	1549	6102#
T45	024434	G	1550	6161#
T46	024640	G	1551	6230#
T47	025044	G	1552	6299#
T48	025250	G	1553	6368#
T49	025454	G	1554	6437#
T5	012112	G	1510	3473#
T50	025660	G	1555	6506#
T51	026064	G	1556	6575#
T52	026270	G	1557	6644#
T53	026474	G	1558	6713#
T54	026702	G	1559	6784#
T55	027106	G	1560	6854#
T56	027312	G	1561	6925#
T57	027516	G	1562	6995#
T58	027722	G	1563	7065#
T59	030126	G	1564	7135#
T6	012246	G	1511	3526#
T60	030332	G	1565	7206#
T61	030536	G	1566	7276#
T62	030742	G	1567	7347#

```
T63      031146 G          1568    7417#
T64      031352 G          1569    7487#
T65      031556 G          1570    7557#
T66      031762 G          1571    7627#
T67      032166 G          1572    7697#
T68      032372 G          1573    7767#
T69      032576 G          1574    7837#
T7       012376 G          1512    3576#
T70      033002 G          1575    7907#
T71      033206 G          1576    7978#
T72      033412 G          1577    8048#
T73      033616 G          1578    8118#
T74      034022 G          1579    8188#
T75      034226 G          1580    8257#
T76      034432 G          1581    8327#
T8       012526 G          1513    3625#
T9       012670 G          1514    3683#
UAM    = 000200 G          1722#
VECTOR   035022            8422    8455#
WMP      034704            8411    8440#
WTYPE    002626            1808#   3098*   3162    3164    3169
X$ALWA= 000000             1368#
X$FALS= 000040             1368#
X$OFFS= 000400             1368#
X$TRUE= 000020             1368#
ZERO     002600            1796#
$BDADR   002634            1811#   2184*   3292*   3321*   3366*   3389*   3430*   3442*   3485*   3502*   3538*   3555*   3588*
                           3605*   3640*   3652*   3697*   3720*   3761*   3785*   3826*   3849*   3891*   3909*   3951*   5463*
                           5469*   5508*   5514*   5568*   5618*   5626*   5674*   5726*   5776*   5847*   5901*   5950*   5965*
                           6012*   6060*   6131*   6193*   6262*   6331*   6400*   6469*   6538*   6607*   6676*   6745*   6816*
                           6886*   6957*   7027*   7097*   7167*   7238*   7308*   7379*   7449*   7519*   7589*   7659*   7729*
                           7799*   7869*   7939*   8010*   8080*   8150*   8220*   8289*   8359*
$BDDAT   002640            1813#   2184*   2499    2516    2533    2551    2569    2587    2605    2623    2640    2657    2684
                           2701    2738    2775    2802    2819    2836    2854    2871    2889    2926    2943    2960    2977
                           3292*   3321*   3366*   3389*   3430*   3442*   3485*   3502*   3538*   3555*   3588*   3605*   3640*
                           3652*   3697*   3720*   3761*   3785*   3826*   3849*   3891*   3909*   3951*   3995*   4025*   4073*
                           4105*   4151*   4181*   4229*   4259*   4311*   4346*   4399*   4434*   4483*   4513*   4561*   4591*
                           4639*   4669*   4717*   4747*   4795*   4825*   4873*   4903*   4951*   4981*   5029*   5059*   5115*
                           5143*   5192*   5223*   5283*   5320*   5383*   5415*   5463*   5469*   5508*   5514*   5568*   5618*
                           5626*   5674*   5726*   5776*   5847*   5901*   5950*   5965*   6012*   6060*   6131*   6193*   6262*
                           6331*   6400*   6469*   6538*   6607*   6676*   6745*   6816*   6886*   6957*   7027*   7097*   7167*
                           7238*   7308*   7379*   7449*   7519*   7589*   7659*   7729*   7799*   7869*   7939*   8010*   8080*
                           8150*   8220*   8289*   8359*
$GDADR   002632            1810#   2550    2568    2586    2604    2622    2997    3192*   3290*   5283*   5320*   5383*   5415*
$GDDAT   002636            1812#   2179*   2181    2184*   2500    2517    2534    2552    2570    2588    2606    2624    2641
                           2658    2685    2702    2739    2776    2803    2820    2837    2855    2872    2890    2927    2944
                           2961    2978    3292*   3321*   3366*   3389*   3424*   3425    3427    3438*   3442*   3485*   3502*
                           3538*   3555*   3588*   3605*   3634*   3635    3637    3648*   3697*   3720*   3761*   3785*   3826*
                           3849*   3891*   3909*   3951*   3995*   4025*   4073*   4103*   4151*   4181*   4229*   4259*   4311*
                           4346*   4399*   4434*   4483*   4513*   4561*   4591*   4639*   4669*   4717*   4747*   4795*   4825*
                           4873*   4903*   4951*   4981*   5029*   5059*   5115*   5143*   5192*   5223*   5278*   5280    5283*
                           5315*   5317    5320*   5378*   5380    5383*   5410*   5412    5415*   5463*   5469*   5508*   5514*
                           5548*   5568*   5618*   5626*   5669*   5671    5715*   5723    5771*   5773    5822*   5844    5896*
                           5897*   5898    5944*   5947    5960*   6006*   6009    6055*   6057    6126*   6128    6188*   6190
                           6257*   6259    6326*   6328    6395*   6397    6464*   6466    6533*   6535    6602*   6604    6671*
                           6673    6740*   6742    6811*   6813    6881*   6883    6952*   6954    7022*   7024    7092*   7094
```

```
                    7162*  7164   7233*  7235   7303*  7305   7374*  7376   7444*  7446   7514*  7516   7584*
                    7586   7654*  7656   7724*  7726   7794*  7796   7864*  7866   7934*  7936   8005*  8007
                    8075*  8077   8145*  8147   8215*  8217   8284*  8286   8354*  8356
$LSTIN= 000000      1376#
$LSTTA= 000000      1377#
$TMPO   002550      1784#  2152*  2155   2174
.       = 040004    1359#  1777#  1779#  1801#  1802#  1803#  1804#  1835#  1881#  1907#  1996#  3023   3284
                    3301   3372   3395   3436   3448   3491   3508   3544   3594   3646   3658   3703   3726
                    3767   3791   3832   3855   3897   3915   3958   4001   4031   4079   4109   4157   4187
                    4235   4265   4317   4352   4405   4440   4489   4519   4567   4597   4645   4675   4723
                    4753   4801   4831   4879   4909   4957   4987   5035   5065   5121   5149   5198   5229
                    5289   5326   5389   5421   5680   5683   5732   5735   5782   5785   6042   6066   6082
                    6137   6199   6209   6268   6278   6337   6347   6406   6416   6475   6485   6544   6554
                    6613   6623   6682   6692   6751   6761   6822   6832   6892   6902   6963   6973   7033
                    7043   7103   7113   7173   7183   7244   7254   7314   7324   7385   7395   7455   7465
                    7525   7535   7595   7605   7665   7675   7735   7745   7805   7815   7875   7885   7945
                    7955   8016   8026   8086   8096   8156   8166   8226   8236   8295   8305   8365   8375
                    8538#
.MSTCL  003156      1990#  3356   3418   3475   3528   3578   3627   3686   3750   3816   3879   3940   3975
                    4054   4132   4210   4288   4376   4464   4542   4620   4698   4776   4854   4932   5010
                    5088   5174   5256   5356   5491   5537   5574   5592   5633   5702   5754   5803   5873
                    5926   5988   6034   6106   6165   6234   6303   6372   6441   6510   6579   6648   6717
                    6788   6858   6929   6999   7069   7139   7210   7280   7351   7421   7491   7561   7631
                    7701   7771   7841   7911   7982   8052   8122   8192   8261   8331
.ROMCL  003244      2005#  2015   2018   2021   2028   2036   2044   2052   2061   2076   2139   2158   2161
                    2165   2171   2176   2206   2218   2221   2230   2233   3986   3989   4016   4019   4064
                    4067   4094   4097   4142   4145   4172   4175   4220   4223   4250   4253   4299   4302
                    4334   4337   4387   4390   4422   4425   4474   4477   4504   4507   4552   4555   4582
                    4585   4630   4633   4660   4663   4708   4711   4738   4741   4786   4789   4816   4819
                    4864   4867   4894   4897   4942   4945   4972   4975   5020   5023   5050   5053   5100
                    5109   5137   5183   5186   5213   5216   5268   5273   5276   5305   5310   5313   5368
                    5373   5376   5405   5408   5456   5501   5555   5611   5662   5666   5712   5717   5720
                    5764   5768   5815   5838   5841   5890   5893   5934   5938   5942   5958   5996   6000
                    6004   6052   6115   6118   6121   6124   6178   6183   6186   6247   6252   6255   6316
                    6321   6324   6385   6390   6393   6454   6459   6462   6523   6528   6531   6592   6597
                    6600   6661   6666   6669   6730   6735   6738   6801   6806   6809   6871   6876   6879
                    6942   6947   6950   7012   7017   7020   7082   7087   7090   7152   7157   7160   7223
                    7228   7231   7293   7298   7301   7364   7369   7372   7434   7439   7442   7504   7509
                    7512   7574   7579   7582   7644   7649   7652   7714   7719   7722   7784   7789   7792
                    7854   7859   7862   7924   7929   7932   7995   8000   8003   8065   8070   8073   8135
                    8140   8143   8205   8210   8213   8274   8279   8282   8344   8349   8352
```

```
BADHEA  1958#  3267  3271  3306  3309  3330  3347  3407  3412  3464  3469  3517  3522  3567  3572
        3617   3621  3673  3678  3738  3743  3803  3808  3867  3872  3925  3932  3964  3969  4043
        4048   4121  4126  4199  4204  4277  4282  4365  4370  4453  4458  4531  4536  4609  4614
        4687   4692  4765  4770  4843  4848  4921  4926  4999  5004  5077  5082  5162  5167  5243
        5248   5345  5350  5433  5437  5479  5483  5524  5529  5579  5584  5638  5642  5690  5694
        5742   5746  5792  5797  5856  5865  5913  5918  5975  5980  6021  6026  6094  6098  6150
        6157   6219  6226  6288  6295  6357  6364  6426  6433  6495  6502  6564  6571  6633  6640
        6702   6709  6773  6780  6843  6850  6914  6921  6984  6991  7054  7061  7124  7131  7194
        7202   7265  7272  7335  7342  7406  7413  7476  7483  7546  7553  7616  7624  7686  7693
        7756   7763  7826  7833  7896  7903  7966  7973  8037  8044  8107  8114  8177  8184  8246
        8253   8316  8323
BCOMPL     1#  1368#  3059  3065  3071
BERROR     1#  1368#  1946#  3994  4024  4072  4102  4150  4180  4228  4258  4310  4345  4398  4433
        4482   4512  4560  4590  4638  4668  4716  4746  4794  4824  4872  4902  4950  4980  5028
        5058   5114  5142  5191  5222
BGNAU      1#  1368#  3251
BGNAUT     1#  1368#  3178
BGNCLN     1#  1368#  3210
BGNDU      1#  1368#  3230
BGNHRD     1#  1368#  8405
BGNHW      1#  1368#  1597
BGNINI     1#  1368#  3041
BGNMOD     1#  1368#  1373
BGNMSG     1#  1368#  2492  2509  2526  2543  2561  2579  2597  2615  2633  2650  2667  2677  2694
        2711   2721  2731  2748  2758  2768  2785  2795  2812  2829  2847  2864  2882  2899  2909
        2919   2936  2953  2970  2987
BGNPRO     1#  1368#  1489
BGNPTA     1#  1368#
BGNRPT     1#  1368#  3017
BGNSEG     1#  1368#  3358  3382  3421  3476  3495  3529  3548  3579  3598  3630  3689  3711  3753
        3775   3818  3842  3882  3902  3981  4010  4059  4088  4137  4166  4215  4244  4293  4327
        4381   4415  4469  4498  4547  4576  4625  4654  4703  4732  4781  4810  4859  4888  4937
        4966   5015  5044  5093  5131  5178  5207  5262  5297  5361  5399  6111  6172  6241  6310
        6379   6448  6517  6586  6655  6724  6795  6865  6936  7006  7076  7146  7217  7287  7358
        7428   7498  7568  7638  7708  7778  7848  7918  7989  8059  8129  8199  8268  8338
BGNSET     1#  1368#
BGNSFT     1#  1368#  8517
BGNSRV     1#  1368#
BGNSUB     1#  1368#  5259
BGNSW      1#  1368#  1629
BGNTST     1#  1368#  3274  3312  3350  3415  3472  3525  3575  3624  3682  3746  3811  3875  3935
        3972   4051  4129  4207  4285  4373  4461  4539  4617  4695  4773  4851  4929  5007  5085
        5170   5251  5353  5440  5486  5532  5587  5645  5697  5749  5800  5868  5921  5983  6029
        6101   6160  6229  6298  6367  6436  6505  6574  6643  6712  6783  6853  6924  6994  7064
        7134   7205  7275  7346  7416  7486  7556  7626  7696  7766  7836  7906  7977  8047  8117
        8187   8256  8326
BNCOMP     1#  1368#  3078  3095
BNERRO     1#  1368#
BREAK      1#  1368#
BRESET     1#  1368#  3212  3233  5444  5565  5647
CKLOOP     1#  1368#
CLOCK      1#  1368#
CLOSE      1#  1368#
CLRVEC     1#  1368#
COMMEN     1#  1368#
DELAY      1#  1368#
```

```
DESCRI      1#   1368#  1759
DEVTYP      1#   1368#  1901
DISPAT      1#   1368#  1503
DISPLA      1#   1368#
DOCLN       1#   1368#
DODU        1#   1368#  3193
DORPT       1#   1368#
ED$CAL   1953#   3268   3272   3307   3310   3331   3348   3408   3413   3465   3470   3518   3523   3568   3573
         3618    3622   3674   3679   3739   3744   3804   3809   3868   3873   3926   3933   3965   3970   4044
         4049    4122   4127   4200   4205   4278   4283   4366   4371   4454   4459   4532   4537   4610   4615
         4688    4693   4766   4771   4844   4849   4922   4927   5000   5005   5078   5083   5163   5168   5244
         5249    5346   5351   5434   5438   5480   5484   5525   5530   5580   5585   5639   5643   5691   5695
         5743    5747   5793   5798   5857   5866   5914   5919   5976   5981   6022   6027   6095   6099   6151
         6158    6220   6227   6289   6296   6358   6365   6427   6434   6496   6503   6565   6572   6634   6641
         6703    6710   6774   6781   6844   6851   6915   6922   6985   6992   7055   7062   7125   7132   7195
         7203    7266   7273   7336   7343   7407   7414   7477   7484   7547   7554   7617   7625   7687   7694
         7757    7764   7827   7834   7897   7904   7967   7974   8038   8045   8108   8115   8178   8185   8247
         8254    8317   8324
ENDAU       1#   1368#  3253
ENDAUT      1#   1368#  3199
ENDCLN      1#   1368#  3215
ENDCOM      1#   1368#
ENDDU       1#   1368#  3235
ENDHRD      1#   1368#  8436
ENDHW       1#   1368#  1715
ENDINI      1#   1368#  3173
ENDMOD      1#   1368#  8532
ENDMSG      1#   1368#  2506   2523   2540   2558   2576   2594   2612   2630   2647   2664   2674   2691   2708
         2718    2728   2745   2755   2765   2782   2792   2809   2826   2844   2861   2879   2896   2906   2916
         2933    2950   2967   2984   3003
ENDPRO      1#   1368#  1494
ENDPTA      1#   1368#
ENDRPT      1#   1368#  3026
ENDSEG      1#   1368#  3377   3400   3451   3492   3509   3545   3559   3595   3609   3661   3707   3731   3771
         3796    3839   3860   3898   3916   4005   4036   4083   4114   4161   4192   4239   4270   4322   4358
         4410    4446   4493   4524   4571   4602   4649   4680   4727   4758   4805   4836   4883   4914   4961
         4992    5039   5070   5126   5155   5202   5236   5293   5331   5394   5426   6138   6204   6273   6342
         6411    6480   6549   6618   6687   6756   6827   6897   6968   7038   7108   7178   7249   7319   7390
         7460    7530   7600   7670   7740   7810   7880   7950   8021   8091   8161   8231   8300   8370
ENDSET      1#   1368#
ENDSFT      1#   1368#  8522
ENDSRV      1#   1368#
ENDSUB      1#   1368#  5338
ENDSW       1#   1368#  1635
ENDTST      1#   1368#  3302   3326   3403   3454   3513   3563   3613   3664   3734   3799   3863   3919   3960
         4039    4117   4195   4273   4361   4449   4527   4605   4683   4761   4839   4917   4995   5073   5158
         5239    5341   5429   5475   5520   5575   5634   5686   5738   5788   5852   5909   5971   6017   6086
         6141    6215   6284   6353   6422   6491   6560   6629   6698   6768   6838   6909   6979   7049   7119
         7189    7260   7330   7401   7471   7541   7611   7681   7751   7821   7891   7961   8032   8102   8172
         8242    8311   8381
EQUALS      1#   1368#  1662
EPRDF       1#   1368#  2184   3292   3321   3366   3389   3430   3442   3485   3502   3538   3555   3588   3605
         3640    3652   3697   3720   3761   3785   3826   3849   3891   3909   3951   3995   4025   4073   4103
         4151    4181   4229   4259   4311   4346   4399   4434   4483   4513   4561   4591   4639   4669   4717
         4747    4795   4825   4873   4903   4951   4981   5029   5059   5115   5143   5192   5223   5283   5320
         5383    5415   5463   5469   5508   5514   5568   5618   5626   5674   5726   5776   5847   5901   5950
```

```
                                                                          I 1
CZDMPD0 M8207 STATIC DIAG #1     MACY11 30A(1052)  30-AUG-82  16:07  PAGE 217
CZDMPD.P11     30-AUG-82 16:03             CROSS REFERENCE TABLE -- MACRO NAMES

          5965    6012    6060    6131    6193    6262    6331    6400    6469    6538    6607    6676    6745    6816    6886
          6957    7027    7097    7167    7238    7308    7379    7449    7519    7589    7659    7729    7799    7869    7939
          8010    8080    8150    8220    8289    8359
ERRHRD       1#   1368#
ERROR        1#   1368#   1931#   2183    3291    3320    3365    3388    3429    3441    3484    3501    3537    3554    3587
          3604    3639    3651    3696    3719    3760    3784    3825    3848    3890    3908    3950    5462    5468    5507
          5513    5567    5617    5625    5673    5725    5775    5846    5900    5949    5964    6011    6059    6130    6192
          6261    6330    6399    6468    6537    6606    6675    6744    6815    6885    6956    7026    7096    7166    7237
          7307    7378    7448    7518    7588    7658    7728    7798    7868    7938    8009    8079    8149    8219    8288
          8358
ERRSF        1#   1368#
ERRSOF       1#   1368#
ERRTBL       1#   1368#
ESCAPE       1#   1368#   3282    3299    3370    3393    3434    3446    3489    3506    3542    3592    3644    3656    3701
          3724    3765    3789    3830    3853    3895    3913    3956    3999    4029    4077    4107    4155    4185    4233
          4263    4315    4350    4403    4438    4487    4517    4565    4595    4643    4673    4721    4751    4799    4829
          4877    4907    4955    4985    5033    5063    5119    5147    5196    5227    5287    5324    5387    5419    5678
          5730    5780    6064    6135    6197    6266    6335    6404    6473    6542    6611    6680    6749    6820    6890
          6961    7031    7101    7171    7242    7312    7383    7453    7523    7593    7663    7733    7803    7873    7943
          8014    8084    8154    8224    8293    8363
EXIT         1#   1368#   3021    5681    5733    5783    6080    6207    6276    6345    6414    6483    6552    6621    6690
          6759    6830    6900    6971    7041    7111    7181    7252    7322    7393    7463    7533    7603    7673    7743
          7813    7883    7953    8024    8094    8164    8234    8303    8373
FEQUAL       1#   1368#
GETBYT       1#   1368#
GETPRI       1#   1368#
GETWOR       1#   1368#
GMANIA       1#   1368#
GMANID       1#   1368#
GMANIL       1#   1368#
GPHARD       1#   1368#   3091
GPRMA        1#   1368#   8415    8420
GPRMD        1#   1368#   8409    8425
GPRML        1#   1368#
HEADER       1#   1368#   1400
INLOOP       1#   1368#
IOSETU       1#   1368#
IOSTAR       1#   1368#
KT11         1#   1368#
K4ONLY    1963#
LASTAD       1#   1368#   8540
MANUAL       1#   1368#
MDT0      2458#   2674    2718    2728    2755    2765    2792    2906    2916
MDT1      2461#   2550    2568    2586    2604    2622
MDT2      2465#   2499    2516    2533    2640    2657    2684    2738    2802    2819    2854    2889    2926    2943    2960
          2977
MDT27     2472#   2836    2871
MDT5      2469#   2701    2775
MEMORY       1#   1368#
MSTCLR    1984#   3355    3417    3474    3527    3577    3626    3685    3749    3815    3878    3939    3974    4053    4131
          4209    4287    4375    4463    4541    4619    4697    4775    4853    4931    5009    5087    5173    5255    5355
          5490    5536    5573    5591    5632    5701    5753    5802    5872    5925    5987    6033    6105    6164    6233
          6302    6371    6440    6509    6578    6647    6716    6787    6857    6928    6998    7068    7138    7209    7279
          7350    7420    7490    7560    7630    7700    7770    7840    7910    7981    8051    8121    8191    8260    8330
MYINT     1972#   3314    3352    3813    3880    3937    3979    4057    4135    4213    4291    4379    4467    4545    4623
          4701    4779    4857    4935    5013    5091    5176    5253    5359    5442    5488    5534    5589    5650    5699
```

```
        5751    5804    5870    5923    5985    6031    6103    6162    6231    6300    6369    6438    6507    6576    6645
        6714    6785    6855    6926    6996    7066    7136    7207    7277    7348    7418    7488    7558    7628    7698
        7768    7838    7908    7979    8049    8119    8189    8258    8328
M$BYTE     1#  1368#   1401#   1407    1408    1409
M$CHEC     1#  1368#   3022#   5682#   5734#   5784#   6081#   6208#   6277#   6346#   6415#   6484#   6553#   6622#   6691#
        6760#   6831#   6901#   6972#   7042#   7112#   7182#   7253#   7323#   7394#   7464#   7534#   7604#   7674#   7744#
        7814#   7884#   7954#   8025#   8095#   8165#   8235#   8304#   8374#
M$CNTO     1#  1368#   8410#   8416#   8421#   8426#
M$COUN     1#  1368#   2493#   2499#   2510#   2516#   2527#   2533#   2544#   2550#   2562#   2568#   2580#   2586#   2598#
        2604#   2616#   2622#   2634#   2640#   2651#   2657#   2668#   2678#   2684#   2695#   2701#   2712#   2722#   2732#
        2738#   2749#   2759#   2769#   2775#   2786#   2796#   2802#   2813#   2819#   2830#   2836#   2848#   2854#   2865#
        2871#   2883#   2889#   2900#   2910#   2920#   2926#   2937#   2943#   2954#   2960#   2971#   2977#   2990#   2997#
M$DATA     1#  1368#   1401#   1410    1412    1414    1416    1418    1420    1422    1424    1426    1428    1430    1432
        1434    1436    1438    1440#   1442    1444    1447    1450    1452    1454    1456    1458    1460    1462    1464
        1466    1468    1470    1472    1474    1476    1478    1480    1482    1484    1760#   1902#
M$DECR     1#  1368#   1495#   1616#   1636#   2506#   2523#   2540#   2558#   2576#   2594#   2612#   2630#   2647#   2664#
        2674#   2691#   2708#   2718#   2728#   2745#   2755#   2765#   2782#   2792#   2809#   2826#   2844#   2861#   2879#
        2896#   2906#   2916#   2933#   2950#   2967#   2984#   3004#   3027#   3174#   3200#   3216#   3236#   3254#   3303#
        3327#   3378#   3401#   3404#   3452#   3455#   3493#   3510#   3514#   3546#   3560#   3564#   3596#   3610#   3614#
        3662#   3665#   3708#   3732#   3735#   3772#   3797#   3800#   3840#   3861#   3864#   3899#   3917#   3920#   3961#
        4006#   4037#   4040#   4084#   4115#   4118#   4162#   4193#   4196#   4240#   4271#   4274#   4323#   4359#   4362#
        4411#   4447#   4450#   4494#   4525#   4528#   4572#   4603#   4606#   4650#   4681#   4684#   4728#   4759#   4762#
        4806#   4837#   4840#   4884#   4915#   4918#   4962#   4993#   4996#   5040#   5071#   5074#   5127#   5156#   5159#
        5203#   5237#   5240#   5294#   5332#   5339#   5342#   5395#   5427#   5430#   5476#   5521#   5576#   5635#   5687#
        5739#   5789#   5853#   5910#   5972#   6018#   6087#   6139#   6142#   6205#   6216#   6274#   6285#   6343#   6354#
        6412#   6423#   6481#   6492#   6550#   6561#   6619#   6630#   6688#   6699#   6757#   6769#   6828#   6839#   6898#
        6910#   6969#   6980#   7039#   7050#   7109#   7120#   7179#   7190#   7250#   7261#   7320#   7331#   7391#   7402#
        7461#   7472#   7531#   7542#   7601#   7612#   7671#   7682#   7741#   7752#   7811#   7822#   7881#   7892#   7951#
        7962#   8022#   8033#   8092#   8103#   8162#   8173#   8232#   8243#   8301#   8312#   8371#   8382#   8437#   8523#
        8533#
M$DEFA     1#  1368#   8410#   8416#   8421#   8426#
M$ENDE     1#  1368#   1616#   1636#   2506#   2523#   2540#   2558#   2576#   2594#   2612#   2630#   2647#   2664#   2674#
        2691#   2708#   2718#   2728#   2745#   2755#   2765#   2782#   2792#   2809#   2826#   2844#   2861#   2879#   2896#
        2906#   2916#   2933#   2950#   2967#   2984#   3004#   3027#   3174#   3200#   3216#   3236#   3254#   3303#   3327#
        3378#   3401#   3404#   3452#   3455#   3493#   3510#   3514#   3546#   3560#   3564#   3596#   3610#   3614#   3662#
        3665#   3708#   3732#   3735#   3772#   3797#   3800#   3840#   3861#   3864#   3899#   3917#   3920#   3961#   4006#
        4037#   4040#   4084#   4115#   4118#   4162#   4193#   4196#   4240#   4271#   4274#   4323#   4359#   4362#   4411#
        4447#   4450#   4494#   4525#   4528#   4572#   4603#   4606#   4650#   4681#   4684#   4728#   4759#   4762#   4806#
        4837#   4840#   4884#   4915#   4918#   4962#   4993#   4996#   5040#   5071#   5074#   5127#   5156#   5159#   5203#
        5237#   5240#   5294#   5332#   5339#   5342#   5395#   5427#   5430#   5476#   5521#   5576#   5635#   5687#   5739#
        5789#   5853#   5910#   5972#   6018#   6087#   6139#   6142#   6205#   6216#   6274#   6285#   6343#   6354#   6412#
        6423#   6481#   6492#   6550#   6561#   6619#   6630#   6688#   6699#   6757#   6769#   6828#   6839#   6898#   6910#
        6969#   6980#   7039#   7050#   7109#   7120#   7179#   7190#   7250#   7261#   7320#   7331#   7391#   7402#   7461#
        7472#   7531#   7542#   7601#   7612#   7671#   7682#   7741#   7752#   7811#   7822#   7881#   7892#   7951#   7962#
        8022#   8033#   8092#   8103#   8162#   8173#   8232#   8243#   8301#   8312#   8371#   8382#   8437#   8523#   8533#
M$ERRI     1#  1368#   2184#   3292#   3321#   3366#   3389#   3430#   3442#   3485#   3502#   3538#   3555#   3588#   3605#
        3640#   3652#   3697#   3720#   3761#   3785#   3826#   3849#   3891#   3909#   3951#   3995#   4025#   4073#   4103#
        4151#   4181#   4229#   4259#   4311#   4346#   4399#   4434#   4483#   4513#   4561#   4591#   4639#   4669#   4717#
        4747#   4795#   4825#   4873#   4903#   4951#   4981#   5029#   5059#   5115#   5143#   5192#   5223#   5283#   5320#
        5383#   5415#   5463#   5469#   5508#   5514#   5568#   5618#   5626#   5674#   5726#   5776#   5847#   5901#   5950#
        5965#   6012#   6060#   6131#   6193#   6262#   6331#   6400#   6469#   6538#   6607#   6676#   6745#   6816#   6886#
        6957#   7027#   7097#   7167#   7238#   7308#   7379#   7449#   7519#   7589#   7659#   7729#   7799#   7869#   7939#
        8010#   8080#   8150#   8220#   8289#   8359#
M$ESCA     1#  1368#   3283#   3284    3300#   3301    3371#   3394#   3435#   3447#   3490#   3507#   3543#   3593#   3645#
        3657#   3702#   3725#   3766#   3790#   3831#   3854#   3896#   3914#   3957#   3958    4000#   4030#   4078#   4108#
        4156#   4186#   4234#   4264#   4316#   4351#   4404#   4439#   4488#   4518#   4566#   4596#   4644#   4674#   4722#
```

```
                4752#   4800#   4830#   4878#   4908#   4956#   4986#   5034#   5064#   5120#   5148#   5197#   5228#   5288#   5325#
                5326    5388#   5420#   5679#   5680    5731#   5732    5781#   5782    6065#   6066    6136#   6198#   6267#   6336#
                6405#   6474#   6543#   6612#   6681#   6750#   6821#   6891#   6962#   7032#   7102#   7172#   7243#   7313#   7384#
                7454#   7524#   7594#   7664#   7734#   7804#   7874#   7944#   8015#   8085#   8155#   8225#   8294#   8364#
M$ESCS             1#   1368#   3283#   3300#   3371#   3372    3394#   3395    3435#   3436    3447#   3448    3490#   3491    3507#
                3508    3543#   3544    3593#   3594    3645#   3646    3657#   3658    3702#   3703    3725#   3726    3766#   3767
                3790#   3791    3831#   3832    3854#   3855    3896#   3897    3914#   3915    3957#   4000#   4001    4030#   4031
                4078#   4079    4108#   4109    4156#   4157    4186#   4187    4234#   4235    4264#   4265    4316#   4317    4351#
                4352    4404#   4405    4439#   4440    4488#   4489    4518#   4519    4566#   4567    4596#   4597    4644#   4645
                4674#   4675    4722#   4723    4752#   4753    4800#   4801    4830#   4831    4878#   4879    4908#   4909    4956#
                4957    4986#   4987    5034#   5035    5064#   5065    5120#   5121    5148#   5149    5197#   5198    5228#   5229
                5288#   5289    5325#   5388#   5389    5420#   5421    5679#   5731#   5781#   6065#   6136#   6137    6198#   6199
                6267#   6268    6336#   6337    6405#   6406    6474#   6475    6543#   6544    6612#   6613    6681#   6682    6750#
                6751    6821#   6822    6891#   6892    6962#   6963    7032#   7033    7102#   7103    7172#   7173    7243#   7244
                7313#   7314    7384#   7385    7454#   7455    7524#   7525    7594#   7595    7664#   7665    7734#   7735    7804#
                7805    7874#   7875    7944#   7945    8015#   8016    8085#   8086    8155#   8156    8225#   8226    8294#   8295
                8364#   8365
M$EXCP             1#   1368#   8410#   8416#   8421#   8426#
M$EXIT             1#   1368#   3022#   5682#   5683    5734#   5735    5784#   5785    6081#   6082    6208#   6209    6277#   6278
                6346#   6347    6415#   6416    6484#   6485    6553#   6554    6622#   6623    6691#   6692    6760#   6761    6831#
                6832    6901#   6902    6972#   6973    7042#   7043    7112#   7113    7182#   7183    7253#   7254    7323#   7324
                7394#   7395    7464#   7465    7534#   7535    7604#   7605    7674#   7675    7744#   7745    7814#   7815    7884#
                7885    7954#   7955    8025#   8026    8095#   8096    8165#   8166    8235#   8236    8304#   8305    8374#   8375
M$EXSE             1#   1368#   3022#   5682#   5734#   5784#   6081#   6208#   6277#   6346#   6415#   6484#   6553#   6622#   6691#
                6760#   6831#   6901#   6972#   7042#   7112#   7182#   7253#   7323#   7394#   7464#   7534#   7604#   7674#   7744#
                7814#   7884#   7954#   8025#   8095#   8165#   8235#   8304#   8374#
M$EXTJ             1#   1368#   3022#   3023    5682#   5734#   5784#   6081#   6208#   6277#   6346#   6415#   6484#   6553#   6622#
                6691#   6760#   6831#   6901#   6972#   7042#   7112#   7182#   7253#   7323#   7394#   7464#   7534#   7604#   7674#
                7744#   7814#   7884#   7954#   8025#   8095#   8165#   8235#   8304#   8374#
M$GEN              1#   1368#   1374#   1401#   1410#   1412#   1414#   1416#   1418#   1420#   1422#   1424#   1426#   1428#   1430#
                1432#   1434#   1436#   1438#   1440#   1442#   1444#   1447#   1450#   1452#   1454#   1456#   1458#   1460#   1462#
                1464#   1466#   1468#   1470#   1472#   1474#   1476#   1478#   1480#   1482#   1484#   1490#   1505#   1599#   1600#
                1616#   1631#   1632#   1636#   1760#   1902#   2492#   2506#   2509#   2523#   2526#   2540#   2543#   2558#   2561#
                2576#   2579#   2594#   2597#   2612#   2615#   2630#   2633#   2647#   2650#   2664#   2667#   2674#   2677#   2691#
                2694#   2708#   2711#   2718#   2721#   2728#   2731#   2745#   2748#   2755#   2758#   2765#   2768#   2782#   2785#
                2792#   2795#   2809#   2812#   2826#   2829#   2844#   2847#   2861#   2864#   2879#   2882#   2896#   2899#   2906#
                2909#   2916#   2919#   2933#   2936#   2950#   2953#   2967#   2970#   2984#   2988#   3004#   3018#   3027#   3042#
                3174#   3179#   3200#   3211#   3216#   3231#   3236#   3252#   3254#   3275#   3303#   3313#   3327#   3351#   3378#
                3401#   3404#   3416#   3452#   3455#   3473#   3493#   3510#   3514#   3526#   3546#   3560#   3564#   3576#   3596#
                3610#   3614#   3625#   3662#   3665#   3683#   3708#   3732#   3735#   3747#   3772#   3797#   3800#   3812#   3840#
                3861#   3864#   3876#   3899#   3917#   3920#   3936#   3961#   3973#   4006#   4037#   4040#   4052#   4084#   4115#
                4118#   4130#   4162#   4193#   4196#   4208#   4240#   4271#   4274#   4286#   4323#   4359#   4362#   4374#   4411#
                4447#   4450#   4462#   4494#   4525#   4528#   4540#   4572#   4603#   4606#   4618#   4650#   4681#   4684#   4696#
                4728#   4759#   4762#   4774#   4806#   4837#   4840#   4852#   4884#   4915#   4918#   4930#   4962#   4993#   4996#
                5008#   5040#   5071#   5074#   5086#   5127#   5156#   5159#   5171#   5203#   5237#   5240#   5252#   5260#   5294#
                5332#   5339#   5342#   5354#   5395#   5427#   5430#   5441#   5476#   5487#   5521#   5533#   5576#   5588#   5635#
                5646#   5687#   5698#   5739#   5750#   5789#   5801#   5853#   5869#   5910#   5922#   5972#   5984#   6018#   6030#
                6087#   6102#   6139#   6142#   6161#   6205#   6216#   6230#   6274#   6285#   6299#   6343#   6354#   6368#   6412#
                6423#   6437#   6481#   6492#   6506#   6550#   6561#   6575#   6619#   6630#   6644#   6688#   6699#   6713#   6757#
                6769#   6784#   6828#   6839#   6854#   6898#   6910#   6925#   6969#   6980#   6995#   7039#   7050#   7065#   7109#
                7120#   7135#   7179#   7190#   7206#   7250#   7261#   7276#   7320#   7331#   7347#   7391#   7402#   7417#   7461#
                7472#   7487#   7531#   7542#   7557#   7601#   7612#   7627#   7671#   7682#   7697#   7741#   7752#   7767#   7811#
                7822#   7837#   7881#   7892#   7907#   7951#   7962#   7978#   8022#   8033#   8048#   8092#   8103#   8118#   8162#
                8173#   8188#   8232#   8243#   8257#   8301#   8312#   8327#   8371#   8382#   8407#   8438#   8519#   8524#   8544#
M$GENB             1#   1368#
M$GETS             1#   1368#   1495#   1616#   1636#   2506#   2523#   2540#   2558#   2576#   2594#   2612#   2630#   2647#   2664#
```

	2674#	2691#	2708#	2718#	2728#	2745#	2755#	2765#	2782#	2792#	2809#	2826#	2844#	2861#	2879#
	2896#	2906#	2916#	2933#	2950#	2967#	2984#	3004#	3027#	3174#	3200#	3216#	3236#	3254#	3303#
	3327#	3372#	3378#	3395#	3401#	3404#	3436#	3448#	3452#	3455#	3491#	3493#	3508#	3510#	3514#
	3544#	3546#	3560#	3564#	3594#	3596#	3610#	3614#	3646#	3658#	3662#	3665#	3703#	3708#	3726#
	3732#	3735#	3767#	3772#	3791#	3797#	3800#	3832#	3840#	3855#	3861#	3864#	3897#	3899#	3915#
	3917#	3920#	3961#	4001#	4006#	4031#	4037#	4040#	4079#	4084#	4109#	4115#	4118#	4157#	4162#
	4187#	4193#	4196#	4235#	4240#	4265#	4271#	4274#	4317#	4323#	4352#	4359#	4362#	4405#	4411#
	4440#	4447#	4450#	4489#	4494#	4519#	4525#	4528#	4567#	4572#	4597#	4603#	4606#	4645#	4650#
	4675#	4681#	4684#	4723#	4728#	4753#	4759#	4762#	4801#	4806#	4831#	4837#	4840#	4879#	4884#
	4909#	4915#	4918#	4957#	4962#	4987#	4993#	4996#	5035#	5040#	5065#	5071#	5074#	5121#	5127#
	5149#	5156#	5159#	5198#	5203#	5229#	5237#	5240#	5289#	5294#	5332#	5339#	5342#	5389#	5395#
	5421#	5427#	5430#	5476#	5521#	5576#	5635#	5687#	5739#	5789#	5853#	5910#	5972#	6018#	6087#
	6137#	6139#	6142#	6199#	6205#	6216#	6268#	6274#	6285#	6337#	6343#	6354#	6406#	6412#	6423#
	6475#	6481#	6492#	6544#	6550#	6561#	6613#	6619#	6630#	6682#	6688#	6699#	6751#	6757#	6769#
	6822#	6828#	6839#	6892#	6898#	6910#	6963#	6969#	6980#	7033#	7039#	7050#	7103#	7109#	7120#
	7173#	7179#	7190#	7244#	7250#	7261#	7314#	7320#	7331#	7385#	7391#	7402#	7455#	7461#	7472#
	7525#	7531#	7542#	7595#	7601#	7612#	7665#	7671#	7682#	7735#	7741#	7752#	7805#	7811#	7822#
	7875#	7881#	7892#	7945#	7951#	7962#	8016#	8022#	8033#	8086#	8092#	8103#	8156#	8162#	8173#
	8226#	8232#	8243#	8295#	8301#	8312#	8365#	8371#	8382#	8437#	8523#	8533#			
M$GETT	1#	1368#	3022#	3283#	3300#	3371#	3372	3394#	3395	3435#	3436	3447#	3448	3490#	3491
	3507#	3508	3543#	3544	3593#	3594	3645#	3646	3657#	3658	3702#	3703	3725#	3726	3766#
	3767	3790#	3791	3831#	3832	3854#	3855	3896#	3897	3914#	3915	3957#	4000#	4001	4030#
	4031	4078#	4079	4108#	4109	4156#	4157	4186#	4187	4234#	4235	4264#	4265	4316#	4317
	4351#	4352	4404#	4405	4439#	4440	4488#	4489	4518#	4519	4566#	4567	4596#	4597	4644#
	4645	4674#	4675	4722#	4723	4752#	4753	4800#	4801	4830#	4831	4878#	4879	4908#	4909
	4956#	4957	4986#	4987	5034#	5035	5064#	5065	5120#	5121	5148#	5149	5197#	5198	5228#
	5229	5288#	5289	5325#	5388#	5389	5420#	5421	5679#	5682#	5731#	5734#	5781#	5784#	6065#
	6081#	6136#	6137	6198#	6199	6208#	6267#	6268	6277#	6336#	6337	6346#	6405#	6406	6415#
	6474#	6475	6484#	6543#	6544	6553#	6612#	6613	6622#	6681#	6682	6691#	6750#	6751	6760#
	6821#	6822	6831#	6891#	6892	6901#	6962#	6963	6972#	7032#	7033	7042#	7102#	7103	7112#
	7172#	7173	7182#	7243#	7244	7253#	7313#	7314	7323#	7384#	7385	7394#	7454#	7455	7464#
	7524#	7525	7534#	7594#	7595	7604#	7664#	7665	7674#	7734#	7735	7744#	7804#	7805	7814#
	7874#	7875	7884#	7944#	7945	7954#	8015#	8016	8025#	8085#	8086	8095#	8155#	8156	8165#
	8225#	8226	8235#	8294#	8295	8304#	8364#	8365	8374#						
M$GNGB	1#	1368#	1374#	1401#	1410#	1412#	1414#	1416#	1418#	1420#	1422#	1424#	1426#	1428#	1430#
	1432#	1434#	1436#	1438#	1440#	1442#	1444#	1447#	1450#	1452#	1454#	1456#	1458#	1460#	1462#
	1464#	1466#	1468#	1470#	1472#	1474#	1476#	1478#	1480#	1482#	1484#	1490#	1504#	1505	1598#
	1599	1600	1630#	1631	1632	1760#	1902#	2492#	2509#	2526#	2543#	2561#	2579#	2597#	2615#
	2633#	2650#	2667#	2677#	2694#	2711#	2721#	2731#	2748#	2758#	2768#	2785#	2795#	2812#	2829#
	2847#	2864#	2882#	2899#	2909#	2919#	2936#	2953#	2970#	2988#	3018#	3042#	3179#	3211#	3231#
	3252#	8406#	8407	8518#	8519	8541#	8544								
M$GNIN	1#	1368#	1401#	1402	1403	1404	1405	1406	1407#	1408#	1409#	1410#	1411	1412#	1413
	1414#	1415	1416#	1417	1418#	1419	1420#	1421	1422#	1423	1424#	1425	1426#	1427	1428#
	1429	1430#	1431	1432#	1433	1434#	1435	1436#	1437	1438#	1439	1440#	1441	1442#	1443
	1444#	1445	1446	1447#	1448	1449#	1450#	1451	1452#	1453	1454#	1455	1456#	1457	1458#
	1459	1460#	1461	1462#	1463	1464#	1465	1466#	1467	1468#	1469	1470#	1471	1472#	1473
	1474#	1475	1476#	1477	1478#	1479	1480#	1481	1482#	1483	1484#	1485	1504#	1506#	1507#
	1508#	1509#	1510#	1511#	1512#	1513#	1514#	1515#	1516#	1517#	1518#	1519#	1520#	1521#	1522#
	1523#	1524#	1525#	1526#	1527#	1528#	1529#	1530#	1531#	1532#	1533#	1534#	1535#	1536#	1537#
	1538#	1539#	1540#	1541#	1542#	1543#	1544#	1545#	1546#	1547#	1548#	1549#	1550#	1551#	1552#
	1553#	1554#	1555#	1556#	1557#	1558#	1559#	1560#	1561#	1562#	1563#	1564#	1565#	1566#	1567#
	1568#	1569#	1570#	1571#	1572#	1573#	1574#	1575#	1576#	1577#	1578#	1579#	1580#	1581#	1598#
	1630#	1760#	1761	1765	1902#	1903	1907	2184#	2185#	2186#	2187#	2493#	2494#	2495#	2496
	2497#	2498	2499#	2500#	2501#	2502#	2503	2504#	2505	2507#	2510#	2511#	2512#	2513	2514#
	2515	2516#	2517#	2518#	2519#	2520	2521#	2522	2524#	2527#	2528#	2529#	2530	2531#	2532
	2533#	2534#	2535#	2536#	2537	2538#	2539	2541#	2544#	2545#	2546#	2547	2548#	2549	2550#

2551# 2552# 2553# 2554# 2555 2556# 2557 2559# 2562# 2563# 2564# 2565 2566# 2567 2568#
2569# 2570# 2571# 2572# 2573 2574# 2575 2577# 2580# 2581# 2582# 2583 2584# 2585 2586#
2587# 2588# 2589# 2590# 2591 2592# 2593 2595# 2598# 2599# 2600# 2601 2602# 2603 2604#
2605# 2606# 2607# 2608# 2609 2610# 2611 2613# 2616# 2617# 2618# 2619 2620# 2621 2622#
2623# 2624# 2625# 2626# 2627 2628# 2629 2631# 2634# 2635# 2636# 2637 2638# 2639 2640#
2641# 2642# 2643# 2644 2645# 2646 2648# 2651# 2652# 2653# 2654 2655# 2656 2657# 2658#
2659# 2660# 2661 2662# 2663 2665# 2668# 2669# 2670# 2671 2672# 2673 2675# 2678# 2679#
2680# 2681 2682# 2683 2684# 2685# 2686# 2687# 2688 2689# 2690 2692# 2695# 2696# 2697#
2698 2699# 2700 2701# 2702# 2703# 2704# 2705 2706# 2707 2709# 2712# 2713# 2714# 2715
2716# 2717 2719# 2722# 2723# 2724# 2725 2726# 2727 2729# 2732# 2733# 2734# 2735 2736#
2737 2738# 2739# 2740# 2741# 2742 2743# 2744 2746# 2749# 2750# 2751# 2752 2753# 2754
2756# 2759# 2760# 2761# 2762 2763# 2764 2766# 2769# 2770# 2771# 2772 2773# 2774 2775#
2776# 2777# 2778# 2779 2780# 2781 2785# 2786# 2787# 2788# 2789 2790# 2791 2793# 2796#
2797# 2798# 2799 2800# 2801 2802# 2803# 2804# 2805# 2806 2807# 2808 2810# 2813# 2814#
2815# 2816 2817# 2818 2819# 2820# 2821# 2822# 2823 2824# 2825 2827# 2830# 2831# 2832#
2833 2834# 2835 2836# 2837# 2838# 2839# 2840# 2841 2842# 2843 2845# 2848# 2849# 2850#
2851 2852# 2853 2854# 2855# 2856# 2857# 2858 2859# 2860 2862# 2865# 2866# 2867# 2868
2869# 2870 2871# 2872# 2873# 2874# 2875# 2876 2877# 2878 2880# 2883# 2884# 2885# 2886
2887# 2888 2889# 2890# 2891# 2892# 2893 2894# 2895 2897# 2900# 2901# 2902# 2903 2904#
2905 2907# 2910# 2911# 2912# 2913 2914# 2915 2917# 2920# 2921# 2922# 2923 2924# 2925
2926# 2927# 2928# 2929# 2930 2931# 2932 2934# 2937# 2938# 2939# 2940 2941# 2942 2943#
2944# 2945# 2946# 2947 2948# 2949 2951# 2954# 2955# 2956# 2957 2958# 2959 2960# 2961#
2962# 2963# 2964 2965# 2966 2968# 2971# 2972# 2973# 2974 2975# 2976 2977# 2978# 2979#
2980# 2981 2982# 2983 2985# 2990# 2991# 2992# 2993 2994# 2995 2997# 2998# 2999# 3000
3001# 3002 3005# 3022# 3023# 3028# 3057# 3058# 3060# 3063# 3064# 3066# 3069# 3070# 3072#
3076# 3077# 3079# 3092# 3093# 3094# 3096# 3175# 3194# 3195# 3201# 3213# 3217# 3234# 3237#
3255# 3283# 3284# 3292# 3293# 3294# 3295# 3300# 3301# 3304# 3321# 3322# 3323# 3324# 3328#
3359# 3366# 3367# 3368# 3369# 3371# 3372# 3379# 3383# 3389# 3390# 3391# 3392# 3394# 3395#
3402# 3405# 3422# 3430# 3431# 3432# 3433# 3435# 3436# 3442# 3443# 3444# 3445# 3447# 3448#
3453# 3456# 3477# 3485# 3486# 3487# 3488# 3490# 3491# 3494# 3496# 3502# 3503# 3504# 3505#
3507# 3508# 3511# 3515# 3530# 3538# 3539# 3540# 3541# 3543# 3544# 3547# 3549# 3555# 3556#
3557# 3558# 3561# 3565# 3580# 3588# 3589# 3590# 3591# 3593# 3594# 3597# 3599# 3605# 3606#
3607# 3608# 3611# 3615# 3631# 3640# 3641# 3642# 3643# 3645# 3646# 3652# 3653# 3654# 3655#
3657# 3658# 3663# 3666# 3690# 3697# 3698# 3699# 3700# 3702# 3703# 3709# 3712# 3720# 3721#
3722# 3723# 3725# 3726# 3733# 3736# 3754# 3761# 3762# 3763# 3764# 3766# 3767# 3773# 3776#
3785# 3786# 3787# 3788# 3790# 3791# 3798# 3801# 3819# 3826# 3827# 3828# 3829# 3831# 3832#
3841# 3843# 3849# 3850# 3851# 3852# 3854# 3855# 3862# 3865# 3883# 3891# 3892# 3893# 3894#
3896# 3897# 3900# 3903# 3909# 3910# 3911# 3912# 3914# 3915# 3918# 3921# 3951# 3952# 3953#
3954# 3957# 3958# 3962# 3982# 3995# 3996# 3997# 3998# 4000# 4001# 4007# 4011# 4025# 4026#
4027# 4028# 4030# 4031# 4038# 4041# 4060# 4073# 4074# 4075# 4076# 4078# 4079# 4085# 4089#
4103# 4104# 4105# 4106# 4108# 4109# 4116# 4119# 4138# 4151# 4152# 4153# 4154# 4156# 4157#
4163# 4167# 4181# 4182# 4183# 4184# 4186# 4187# 4194# 4197# 4216# 4229# 4230# 4231# 4232#
4234# 4235# 4241# 4245# 4259# 4260# 4261# 4262# 4264# 4265# 4272# 4275# 4294# 4311# 4312#
4313# 4314# 4316# 4317# 4324# 4328# 4346# 4347# 4348# 4349# 4351# 4352# 4360# 4363# 4382#
4399# 4400# 4401# 4402# 4404# 4405# 4412# 4416# 4434# 4435# 4436# 4437# 4439# 4440# 4448#
4451# 4470# 4483# 4484# 4485# 4486# 4488# 4489# 4495# 4499# 4513# 4514# 4515# 4516# 4518#
4519# 4526# 4529# 4548# 4561# 4562# 4563# 4564# 4566# 4567# 4573# 4577# 4591# 4592# 4593#
4594# 4596# 4597# 4604# 4607# 4626# 4639# 4640# 4641# 4642# 4644# 4645# 4651# 4655# 4669#
4670# 4671# 4672# 4674# 4675# 4682# 4685# 4704# 4717# 4718# 4719# 4720# 4722# 4723# 4729#
4733# 4747# 4748# 4749# 4750# 4752# 4753# 4760# 4763# 4782# 4795# 4796# 4797# 4798# 4800#
4801# 4807# 4811# 4825# 4826# 4827# 4828# 4830# 4831# 4838# 4841# 4860# 4873# 4874# 4875#
4876# 4878# 4879# 4885# 4889# 4903# 4904# 4905# 4906# 4908# 4909# 4916# 4919# 4938# 4951#
4952# 4953# 4954# 4956# 4957# 4963# 4967# 4981# 4982# 4983# 4984# 4986# 4987# 4994# 4997#
5016# 5029# 5030# 5031# 5032# 5034# 5035# 5041# 5045# 5059# 5060# 5061# 5062# 5064# 5065#
5072# 5075# 5094# 5115# 5116# 5117# 5118# 5120# 5121# 5128# 5132# 5143# 5144# 5145# 5146#
5148# 5149# 5157# 5160# 5179# 5192# 5193# 5194# 5195# 5197# 5198# 5204# 5208# 5223# 5224#

	5225#	5226#	5228#	5229#	5238#	5241#	5261#	5263#	5283#	5284#	5285#	5286#	5288#	5289#	5295#
	5298#	5320#	5321#	5322#	5323#	5325#	5326#	5333#	5340#	5343#	5362#	5383#	5384#	5385#	5386#
	5388#	5389#	5396#	5400#	5415#	5416#	5417#	5418#	5420#	5421#	5428#	5431#	5445#	5452#	5453#
	5459#	5460#	5463#	5464#	5465#	5466#	5469#	5470#	5471#	5472#	5477#	5497#	5498#	5504#	5505#
	5508#	5509#	5510#	5511#	5514#	5515#	5516#	5517#	5522#	5540#	5541#	5558#	5559#	5566#	5568#
	5569#	5570#	5571#	5577#	5595#	5596#	5614#	5615#	5618#	5619#	5620#	5621#	5626#	5627#	5628#
	5629#	5636#	5648#	5674#	5675#	5676#	5677#	5679#	5680#	5682#	5683#	5688#	5726#	5727#	5728#
	5729#	5731#	5732#	5734#	5735#	5740#	5776#	5777#	5778#	5779#	5781#	5782#	5784#	5785#	5790#
	5847#	5848#	5849#	5850#	5854#	5901#	5902#	5903#	5904#	5911#	5950#	5951#	5952#	5953#	5965#
	5966#	5967#	5968#	5973#	6012#	6013#	6014#	6015#	6019#	6060#	6061#	6062#	6063#	6065#	6066#
	6081#	6082#	6086#	6112#	6131#	6132#	6133#	6134#	6136#	6137#	6140#	6143#	6173#	6193#	6194#
	6195#	6196#	6198#	6199#	6206#	6208#	6209#	6217#	6242#	6262#	6263#	6264#	6265#	6267#	6268#
	6275#	6277#	6278#	6286#	6311#	6331#	6332#	6333#	6334#	6336#	6337#	6344#	6346#	6347#	6355#
	6380#	6400#	6401#	6402#	6403#	6405#	6406#	6413#	6415#	6416#	6424#	6449#	6469#	6470#	6471#
	6472#	6474#	6475#	6482#	6484#	6485#	6493#	6518#	6538#	6539#	6540#	6541#	6543#	6544#	6551#
	6553#	6554#	6562#	6587#	6607#	6608#	6609#	6610#	6612#	6613#	6620#	6622#	6623#	6631#	6656#
	6676#	6677#	6678#	6679#	6681#	6682#	6689#	6691#	6692#	6700#	6725#	6745#	6746#	6747#	6748#
	6750#	6751#	6758#	6760#	6761#	6770#	6796#	6816#	6817#	6818#	6819#	6821#	6822#	6829#	6831#
	6832#	6840#	6866#	6886#	6887#	6888#	6889#	6891#	6892#	6899#	6901#	6902#	6911#	6937#	6957#
	6958#	6959#	6960#	6962#	6963#	6970#	6972#	6973#	6981#	7007#	7027#	7028#	7029#	7030#	7032#
	7033#	7040#	7042#	7043#	7051#	7077#	7097#	7098#	7099#	7100#	7102#	7103#	7110#	7112#	7113#
	7121#	7147#	7167#	7168#	7169#	7170#	7172#	7173#	7180#	7182#	7183#	7191#	7218#	7238#	7239#
	7240#	7241#	7243#	7244#	7251#	7253#	7254#	7262#	7288#	7308#	7309#	7310#	7311#	7313#	7314#
	7321#	7323#	7324#	7332#	7359#	7379#	7380#	7381#	7382#	7384#	7385#	7392#	7394#	7395#	7403#
	7429#	7449#	7450#	7451#	7452#	7454#	7455#	7462#	7464#	7465#	7473#	7499#	7519#	7520#	7521#
	7522#	7524#	7525#	7532#	7534#	7535#	7543#	7569#	7589#	7590#	7591#	7592#	7594#	7595#	7602#
	7604#	7605#	7613#	7639#	7659#	7660#	7661#	7662#	7664#	7665#	7672#	7674#	7675#	7683#	7709#
	7729#	7730#	7731#	7732#	7734#	7735#	7742#	7744#	7745#	7753#	7779#	7799#	7800#	7801#	7802#
	7804#	7805#	7812#	7814#	7815#	7823#	7849#	7869#	7870#	7871#	7872#	7874#	7875#	7882#	7884#
	7885#	7893#	7919#	7939#	7940#	7941#	7942#	7944#	7945#	7952#	7954#	7955#	7963#	7990#	8010#
	8011#	8012#	8013#	8015#	8016#	8023#	8025#	8026#	8034#	8060#	8080#	8081#	8082#	8083#	8085#
	8086#	8093#	8095#	8096#	8104#	8130#	8150#	8151#	8152#	8153#	8155#	8156#	8163#	8165#	8166#
	8174#	8200#	8220#	8221#	8222#	8223#	8225#	8226#	8233#	8235#	8236#	8244#	8269#	8289#	8290#
	8291#	8292#	8294#	8295#	8302#	8304#	8305#	8313#	8339#	8359#	8360#	8361#	8362#	8364#	8365#
	8372#	8374#	8375#	8383#	8406#	8410#	8411	8412	8413	8414	8416#	8417	8418	8419	8421#
	8422	8423	8424	8426#	8427	8428	8429	8430	8437#	8518#	8523#	8541#	8542#	8543#	
MSGNLS	1#	1368#	3378#	3401#	3452#	3493#	3510#	3546#	3560#	3596#	3610#	3662#	3708#	3732#	3772#
	3797#	3840#	3861#	3899#	3917#	4006#	4037#	4084#	4115#	4162#	4193#	4240#	4271#	4323#	4359#
	4411#	4447#	4494#	4525#	4572#	4603#	4650#	4681#	4728#	4759#	4806#	4837#	4884#	4915#	4962#
	4993#	5040#	5071#	5127#	5156#	5203#	5237#	5294#	5332#	5395#	5427#	6139#	6205#	6274#	6343#
	6412#	6481#	6550#	6619#	6688#	6757#	6828#	6898#	6969#	7039#	7109#	7179#	7250#	7320#	7391#
	7461#	7531#	7601#	7671#	7741#	7811#	7881#	7951#	8022#	8092#	8162#	8232#	8301#	8371#	
MSGNSU	1#	1368#	5260#												
MSGNTA	1#	1368#	1616#	1676#	2506#	2523#	2540#	2558#	2576#	2594#	2612#	2630#	2647#	2664#	2674#
	2691#	2708#	2718#	2728#	2745#	2755#	2765#	2782#	2792#	2809#	2826#	2844#	2861#	2879#	2896#
	2906#	2916#	2933#	2950#	2967#	2984#	3004#	3027#	3174#	3200#	3216#	3236#	3254#	3303#	3327#
	3404#	3455#	3514#	3564#	3614#	3665#	3735#	3800#	3864#	3920#	3961#	4040#	4118#	4196#	4274#
	4362#	4450#	4528#	4606#	4684#	4762#	4840#	4918#	4996#	5074#	5159#	5240#	5339#	5342#	5430#
	5476#	5521#	5576#	5635#	5687#	5739#	5789#	5853#	5910#	5972#	6018#	6087#	6142#	6216#	6285#
	6354#	6423#	6492#	6561#	6630#	6699#	6769#	6839#	6910#	6980#	7050#	7120#	7190#	7261#	7331#
	7402#	7472#	7542#	7612#	7682#	7752#	7822#	7892#	7962#	8033#	8103#	8173#	8243#	8312#	8382#
	8437#	8438	8523#	8524											
MSGNTE	1#	1368#	3275#	3313#	3351#	3416#	3473#	3526#	3576#	3625#	3683#	3747#	3812#	3876#	3936#
	3973#	4052#	4130#	4208#	4286#	4374#	4462#	4540#	4618#	4696#	4774#	4852#	4930#	5008#	5086#
	5171#	5252#	5354#	5441#	5487#	5533#	5588#	5646#	5698#	5750#	5801#	5869#	5922#	5984#	6030#
	6102#	6161#	6230#	6299#	6368#	6437#	6506#	6575#	6644#	6713#	6784#	6854#	6925#	6995#	7065#

	7135#	7206#	7276#	7347#	7417#	7487#	7557#	7627#	7697#	7767#	7837#	7907#	7978#	8048#	8118#
	8188#	8257#	8327#												
M$HAPT	1#	1368#	1401#												
M$HNAP	1#	1368#	1401#	1440											
M$INCR	1#	1368#	1374#	1490#	1598#	1630#	2184#	2492#	2497#	2504#	2507#	2509#	2514#	2521#	2524#
	2526#	2531#	2538#	2541#	2543#	2548#	2556#	2559#	2561#	2566#	2574#	2577#	2579#	2584#	2592#
	2595#	2597#	2602#	2610#	2613#	2615#	2620#	2628#	2631#	2633#	2638#	2645#	2648#	2650#	2655#
	2662#	2665#	2667#	2672#	2675#	2677#	2682#	2689#	2692#	2694#	2699#	2706#	2709#	2711#	2716#
	2719#	2721#	2726#	2729#	2731#	2736#	2743#	2746#	2748#	2753#	2756#	2758#	2763#	2766#	2768#
	2773#	2780#	2783#	2785#	2790#	2793#	2795#	2800#	2807#	2810#	2812#	2817#	2824#	2827#	2829#
	2834#	2842#	2845#	2847#	2852#	2859#	2862#	2864#	2869#	2877#	2880#	2882#	2887#	2894#	2897#
	2899#	2904#	2907#	2909#	2914#	2917#	2919#	2924#	2931#	2934#	2936#	2941#	2948#	2951#	2953#
	2958#	2965#	2968#	2970#	2975#	2982#	2985#	2988#	2994#	3001#	3005#	3018#	3028#	3042#	3058#
	3064#	3070#	3077#	3093#	3175#	3179#	3195#	3201#	3211#	3213#	3217#	3231#	3234#	3237#	3252#
	3255#	3275#	3276#	3283#	3292#	3300#	3304#	3313#	3314#	3321#	3328#	3351#	3352#	3359#	3366#
	3371#	3379#	3383#	3389#	3394#	3402#	3405#	3416#	3417#	3422#	3430#	3435#	3442#	3447#	3453#
	3456#	3473#	3474#	3477#	3485#	3490#	3494#	3496#	3502#	3507#	3511#	3515#	3526#	3527#	3530#
	3538#	3543#	3547#	3549#	3555#	3561#	3565#	3576#	3577#	3580#	3588#	3593#	3597#	3599#	3605#
	3611#	3615#	3625#	3626#	3631#	3640#	3645#	3652#	3657#	3663#	3666#	3683#	3684#	3690#	3697#
	3702#	3709#	3712#	3720#	3725#	3733#	3736#	3747#	3748#	3754#	3761#	3766#	3773#	3776#	3785#
	3790#	3798#	3801#	3812#	3813#	3819#	3826#	3831#	3841#	3843#	3849#	3854#	3862#	3865#	3876#
	3877#	3883#	3891#	3896#	3900#	3903#	3909#	3914#	3918#	3921#	3936#	3937#	3951#	3957#	3962#
	3973#	3974#	3982#	3995#	4000#	4007#	4011#	4025#	4030#	4038#	4041#	4052#	4053#	4060#	4073#
	4078#	4085#	4089#	4103#	4108#	4116#	4119#	4130#	4131#	4138#	4151#	4156#	4163#	4167#	4181#
	4186#	4194#	4197#	4208#	4209.	4216#	4229#	4234#	4241#	4245#	4259#	4264#	4272#	4275#	4286#
	4287#	4294#	4311#	4316#	4324#	4328#	4346#	4351#	4360#	4363#	4374#	4375#	4382#	4399#	4404#
	4412#	4416#	4434#	4439#	4448#	4451#	4462#	4463#	4470#	4483#	4488#	4495#	4499#	4513#	4518#
	4526#	4529#	4540#	4541#	4548#	4561#	4566#	4573#	4577#	4591#	4596#	4604#	4607#	4618#	4619#
	4626#	4639#	4644#	4651#	4655#	4669#	4674#	4682#	4685#	4696#	4697#	4704#	4717#	4722#	4729#
	4733#	4747#	4752#	4760#	4763#	4774#	4775#	4782#	4795#	4800#	4807#	4811#	4825#	4830#	4838#
	4841#	4852#	4853#	4860#	4873#	4878#	4885#	4889#	4903#	4908#	4916#	4919#	4930#	4931#	4938#
	4951#	4956#	4963#	4967#	4981#	4986#	4994#	4997#	5008#	5009#	5016#	5029#	5034#	5041#	5045#
	5059#	5064#	5072#	5075#	5086#	5087#	5094#	5115#	5120#	5128#	5132#	5143#	5148#	5157#	5160#
	5171#	5172#	5179#	5192#	5197#	5204#	5208#	5223#	5228#	5238#	5241#	5252#	5253#	5260#	5261#
	5263#	5283#	5288#	5295#	5298#	5320#	5325#	5333#	5340#	5343#	5354#	5355#	5362#	5383#	5388#
	5396#	5400#	5415#	5420#	5428#	5431#	5441#	5442#	5445#	5453#	5460#	5463#	5469#	5477#	5487#
	5488#	5498#	5505#	5508#	5514#	5522#	5533#	5534#	5541#	5559#	5566#	5568#	5577#	5588#	5589#
	5596#	5615#	5618#	5626#	5636#	5646#	5647#	5648#	5674#	5679#	5682#	5688#	5698#	5699#	5726#
	5731#	5734#	5740#	5750#	5751#	5776#	5781#	5784#	5790#	5801#	5802#	5847#	5854#	5869#	5870#
	5901#	5911#	5922#	5923#	5950#	5965#	5973#	5984#	5985#	6012#	6019#	6030#	6031#	6060#	6065#
	6081#	6088#	6102#	6103#	6112#	6131#	6136#	6140#	6143#	6161#	6162#	6173#	6193#	6198#	6206#
	6208#	6217#	6230#	6231#	6242#	6262#	6267#	6275#	6277#	6286#	6299#	6300#	6311#	6331#	6336#
	6344#	6346#	6355#	6368#	6369#	6380#	6400#	6405#	6413#	6415#	6424#	6437#	6438#	6449#	6469#
	6474#	6482#	6484#	6493#	6506#	6507#	6518#	6538#	6543#	6551#	6553#	6562#	6575#	6576#	6587#
	6607#	6612#	6620#	6622#	6631#	6644#	6645#	6656#	6676#	6681#	6689#	6691#	6700#	6713#	6714#
	6725#	6745#	6750#	6758#	6760#	6770#	6784#	6785#	6796#	6816#	6821#	6829#	6831#	6840#	6854#
	6855#	6866#	6886#	6891#	6899#	6901#	6911#	6925#	6926#	6937#	6957#	6962#	6970#	6972#	6981#
	6995#	6996#	7007#	7027#	7032#	7040#	7042#	7051#	7065#	7066#	7077#	7097#	7102#	7110#	7112#
	7121#	7135#	7136#	7147#	7167#	7172#	7180#	7182#	7191#	7206#	7207#	7218#	7238#	7243#	7251#
	7253#	7262#	7276#	7277#	7288#	7308#	7313#	7321#	7323#	7332#	7347#	7348#	7359#	7379#	7384#
	7392#	7394#	7403#	7417#	7418#	7429#	7449#	7454#	7462#	7464#	7473#	7487#	7488#	7499#	7519#
	7524#	7532#	7534#	7543#	7557#	7558#	7569#	7589#	7594#	7602#	7604#	7613#	7627#	7628#	7639#
	7659#	7664#	7672#	7674#	7683#	7697#	7698#	7709#	7729#	7734#	7742#	7744#	7753#	7767#	7768#
	7779#	7799#	7804#	7812#	7814#	7823#	7837#	7838#	7849#	7869#	7874#	7882#	7884#	7893#	7907#
	7908#	7919#	7939#	7944#	7952#	7954#	7963#	7978#	7979#	7990#	8010#	8015#	8023#	8025#	8034#
	8048#	8049#	8060#	8080#	8085#	8093#	8095#	8104#	8118#	8119#	8130#	8150#	8155#	8163#	8165#

```
                2624    2625    2626    2634#   2635    2636    2640#   2641    2642    2643    2651#   2652    2653    2657#   2658
                2659    2660    2668#   2669    2670    2678#   2679    2680    2684#   2685    2686    2687    2695#   2696    2697
                2701#   2702    2703    2704    2712#   2713    2714    2722#   2723    2724    2732#   2733    2734    2738#   2739
                2740    2741    2749#   2750    2751    2759#   2760    2761    2769#   2770    2771    2775#   2776    2777    2778
                2786#   2787    2788    2796#   2797    2798    2802#   2803    2804    2805    2813#   2814    2815    2819#   2820
                2821    2822    2830#   2831    2832    2836#   2837    2838    2839    2840    2848#   2849    2850    2854#   2855
                2856    2857    2865#   2866    2867    2871#   2872    2873    2874    2875    2883#   2884    2885    2889#   2890
                2891    2892    2900#   2901    2902    2910#   2911    2912    2920#   2921    2922    2926#   2927    2928    2929
                2937#   2938    2939    2943#   2944    2945    2946    2954#   2955    2956    2960#   2961    2962    2963    2971#
                2972    2973    2977#   2978    2979    2980    2990#   2991    2992    2997#   2998    2999
M$RADI             1#   1368#   8410#   8416#   8421#   8426#
M$RBRO             1#   1368#
M$RNRO             1#   1368#   3092#   3094
M$SETS             1#   1368#   1374#   1490#   1598#   1630#   2492#   2509#   2526#   2543#   2561#   2579#   2597#   2615#   2633#
                2650#   2667#   2677#   2694#   2711#   2721#   2731#   2748#   2758#   2768#   2785#   2795#   2812#   2829#   2847#
                2864#   2882#   2899#   2909#   2919#   2936#   2953#   2970#   2988#   3018#   3042#   3179#   3211#   3231#   3252#
                3276#   3314#   3352#   3359#   3383#   3417#   3422#   3474#   3477#   3496#   3527#   3530#   3549#   3577#   3580#
                3599#   3626#   3631#   3684#   3690#   3712#   3748#   3754#   3776#   3813#   3819#   3843#   3877#   3883#   3903#
                3937#   3974#   3982#   4011#   4053#   4060#   4089#   4131#   4138#   4167#   4209#   4216#   4245#   4287#   4294#
                4328#   4375#   4382#   4416#   4463#   4470#   4499#   4541#   4548#   4577#   4619#   4626#   4655#   4697#   4704#
                4733#   4775#   4782#   4811#   4853#   4860#   4889#   4931#   4938#   4967#   5009#   5016#   5045#   5087#   5094#
                5132#   5172#   5179#   5208#   5253#   5261#   5263#   5298#   5355#   5362#   5400#   5442#   5488#   5534#   5589#
                5647#   5699#   5751#   5802#   5870#   5923#   5985#   6031#   6103#   6112#   6162#   6173#   6231#   6242#   6300#
                6311#   6369#   6380#   6438#   6449#   6507#   6518#   6576#   6587#   6645#   6656#   6714#   6725#   6785#   6796#
                6855#   6866#   6926#   6937#   6996#   7007#   7066#   7077#   7136#   7147#   7207#   7218#   7277#   7288#   7348#
                7359#   7418#   7429#   7488#   7499#   7558#   7569#   7628#   7639#   7698#   7709#   7768#   7779#   7838#   7849#
                7908#   7919#   7979#   7990#   8049#   8060#   8119#   8130#   8189#   8200#   8258#   8269#   8328#   8339#   8406#
                8518#
M$STAR             1#   1368#
M$SVC              1#   1368#   2184    2493#   2497    2499#   2504    2506#   2507    2510#   2514    2516#   2521    2523#   2524
                2527#   2531    2533#   2538    2540#   2541    2544#   2548    2550#   2556    2558#   2559    2562#   2566    2568#
                2574    2576#   2577    2580#   2584    2586#   2592    2594#   2595    2598#   2602    2604#   2610    2612#   2613
                2616#   2620    2622#   2628    2630#   2631    2634#   2638    2640#   2645    2647#   2648    2651#   2655    2657#
                2662    2664#   2665    2668#   2672    2674#   2675    2678#   2682    2684#   2689    2691#   2692    2695#   2699
                2701#   2706    2708#   2709    2712#   2716    2718#   2719    2722#   2726    2728#   2729    2732#   2736    2738#
                2743    2745#   2746    2749#   2753    2755#   2756    2759#   2763    2765#   2766    2769#   2773    2775#   2780
                2782#   2783    2786#   2790    2792#   2793    2796#   2800    2802#   2807    2809#   2810    2813#   2817    2819#
                2824    2826#   2827    2830#   2834    2836#   2842    2844#   2845    2848#   2852    2854#   2859    2861#   2862
                2865#   2869    2871#   2877    2879#   2880    2883#   2887    2889#   2894    2896#   2897    2900#   2904    2906#
                2907    2910#   2914    2916#   2917    2920#   2924    2926#   2931    2933#   2934    2937#   2941    2943#   2948
                2950#   2951    2954#   2958    2960#   2965    2967#   2968    2971#   2975    2977#   2982    2984#   2985    2990#
                2994    2997#   3001    3004#   3005    3022#   3027#   3028    3057#   3058    3063#   3064    3069#   3070    3076#
                3077    3092#   3093    3174#   3175    3194#   3195    3200#   3201    3213#   3216#   3217    3234#   3236#   3237
                3254#   3255    3283#   3292    3300#   3303#   3304    3321    3327#   3328    3359#   3366    3371#   3378#   3379
                3383#   3389    3394#   3401#   3402    3404#   3405    3422#   3430    3435#   3442    3447#   3452#   3453    3455#
                3456    3477#   3485    3490#   3493#   3494    3496#   3502    3507#   3510#   3511    3514#   3515    3530#   3538
                3543#   3546#   3547    3549#   3555    3560#   3561    3564#   3565    3580#   3588    3593#   3596#   3597    3599#
                3605    3610#   3611    3614#   3615    3631#   3640    3645#   3652    3657#   3662#   3663    3665#   3666    3690#
                3697    3702#   3708#   3709    3712#   3720    3725#   3732#   3733    3735#   3736    3754#   3761    3766#   3772#
                3773    3776#   3785    3790#   3797#   3798    3800#   3801    3819#   3826    3831#   3840#   3841    3843#   3849
                3854#   3861#   3862    3864#   3865    3883#   3891    3896#   3899#   3900    3903#   3909    3914#   3917#   3918
                3920#   3921    3951    3957#   3961#   3962    3982#   3995    4000#   4006#   4007    4011#   4025    4030#   4037#
                4038    4040#   4041    4060#   4073    4078#   4084#   4085    4089#   4103    4108#   4115#   4116    4118#   4119
                4138#   4151    4156#   4162#   4163    4167#   4181    4186#   4193#   4194    4196#   4197    4216#   4229    4234#
                4240#   4241    4245#   4259    4264#   4271#   4272    4274#   4275    4294#   4311    4316#   4323#   4324    4328#
                4346    4351#   4359#   4360    4362#   4363    4382#   4399    4404#   4411#   4412    4416#   4434    4439#   4447#
```

```
                4448    4450#   4451    4470#   4483    4488#   4494#   4495    4499#   4513    4518#   4525#   4526    4528#   4529
                4548#   4561    4566#   4572#   4573    4577#   4591    4596#   4603#   4604    4606#   4607    4626#   4639    4644#
                4650#   4651    4655#   4669    4674#   4681#   4682    4684#   4685    4704#   4717    4722#   4728#   4729    4733#
                4747    4752#   4759#   4760    4762#   4763    4782#   4795    4800#   4806#   4807    4811#   4825    4830#   4837#
                4838    4840#   4841    4860#   4873    4878#   4884#   4885    4889#   4903    4908#   4915#   4916    4918#   4919
                4938#   4951    4956#   4962#   4963    4967#   4981    4986#   4993#   4994    4996#   4997    5016#   5029    5034#
                5040#   5041    5045#   5059    5064#   5071#   5072    5074#   5075    5094#   5115    5120#   5127#   5128    5132#
                5143    5148#   5156#   5157    5159#   5160    5179#   5192    5197#   5203#   5204    5208#   5223    5228#   5237#
                5238    5240#   5241    5260#   5261    5263#   5283    5288#   5294#   5295    5298#   5320    5325#   5332#   5333
                5339#   5340    5342#   5343    5362#   5383    5388#   5395#   5396    5400#   5415    5420#   5427#   5428    5430#
                5431    5445#   5452#   5453    5459#   5460    5463    5469    5476#   5477    5497#   5498    5504#   5505    5508
                5514    5521#   5522    5540#   5541    5558#   5559    5566#   5568    5576#   5577    5595#   5596    5614#   5615
                5618    5626    5635#   5636    5648#   5674    5679#   5682#   5687#   5688    5726    5731#   5734#   5739#   5740
                5776    5781#   5784#   5789#   5790    5847    5853#   5854    5901    5910#   5911    5950    5965    5972#   5973
                6012    6018#   6019    6060    6065#   6081#   6087#   6088    6112#   6131    6136#   6139#   6140    6142#   6143
                6173#   6193    6198#   6205#   6206    6208#   6216#   6217    6242#   6262    6267#   6274#   6275    6277#   6285#
                6286    6311#   6331    6336#   6343#   6344    6346#   6354#   6355    6380#   6400    6405#   6412#   6413    6415#
                6423#   6424    6449#   6469    6474#   6481#   6482    6484#   6492#   6493    6518#   6538    6543#   6550#   6551
                6553#   6561#   6562    6587#   6607    6612#   6619#   6620    6622#   6630#   6631    6656#   6676    6681#   6688#
                6689    6691#   6699#   6700    6725#   6745    6750#   6757#   6758    6760#   6769#   6770    6796#   6816    6821#
                6828#   6829    6831#   6839#   6840    6866#   6886    6891#   6898#   6899    6901#   6910#   6911    6937#   6957
                6962#   6969#   6970    6972#   6980#   6981    7007#   7027    7032#   7039#   7040    7042#   7050#   7051    7077#
                7097    7102#   7109#   7110    7112#   7120#   7121    7147#   7167    7172#   7179#   7180    7182#   7190#   7191
                7218#   7238    7243#   7250#   7251    7253#   7261#   7262    7288#   7308    7313#   7320#   7321    7323#   7331#
                7332    7359#   7379    7384#   7391#   7392    7394#   7402#   7403    7429#   7449    7454#   7461#   7462    7464#
                7472#   7473    7499#   7519    7524#   7531#   7532    7534#   7542#   7543    7569#   7589    7594#   7601#   7602
                7604#   7612#   7613    7639#   7659    7664#   7671#   7672    7674#   7682#   7683    7709#   7729    7734#   7741#
                7742    7744#   7752#   7753    7779#   7799    7804#   7811#   7812    7814#   7822#   7823    7849#   7869    7874#
                7881#   7882    7884#   7892#   7893    7919#   7939    7944#   7951#   7952    7954#   7962#   7963    7990#   8010
                8015#   8022#   8023    8025#   8033#   8034    8060#   8080    8085#   8092#   8093    8095#   8103#   8104    8130#
                8150    8155#   8162#   8163    8165#   8173#   8174    8200#   8220    8225#   8232#   8233    8235#   8243#   8244
                8269#   8289    8294#   8301#   8302    8304#   8312#   8313    8339#   8359    8364#   8371#   8372    8374#   8382#
                8383
M$TLAB            1#    1368#   2184#   2497#   2504#   2507#   2514#   2521#   2524#   2531#   2538#   2541#   2548#   2556#   2559#
                2566#   2574#   2577#   2584#   2592#   2595#   2602#   2610#   2613#   2620#   2628#   2631#   2638#   2645#   2648#
                2655#   2662#   2665#   2672#   2675#   2682#   2689#   2692#   2699#   2706#   2709#   2716#   2719#   2726#   2729#
                2736#   2743#   2746#   2753#   2756#   2763#   2766#   2773#   2780#   2783#   2790#   2793#   2800#   2807#   2810#
                2817#   2824#   2827#   2834#   2842#   2845#   2852#   2859#   2862#   2869#   2877#   2880#   2887#   2894#   2897#
                2904#   2907#   2914#   2917#   2924#   2931#   2934#   2941#   2948#   2951#   2958#   2965#   2968#   2975#   2982#
                2985#   2994#   3001#   3005#   3028#   3058#   3064#   3070#   3077#   3093#   3175#   3195#   3201#   3213#   3217#
                3234#   3237#   3255#   3283#   3292#   3300#   3304#   3321#   3328#   3359#   3366#   3371#   3379#   3383#   3389#
                3394#   3402#   3405#   3422#   3430#   3435#   3442#   3447#   3453#   3456#   3477#   3485#   3490#   3494#   3496#
                3502#   3507#   3511#   3515#   3530#   3538#   3543#   3547#   3549#   3555#   3561#   3565#   3580#   3588#   3593#
                3597#   3599#   3605#   3611#   3615#   3631#   3640#   3645#   3652#   3657#   3663#   3666#   3690#   3697#   3702#
                3709#   3712#   3720#   3725#   3733#   3736#   3754#   3761#   3766#   3773#   3776#   3785#   3790#   3798#   3801#
                3819#   3826#   3831#   3841#   3843#   3849#   3854#   3862#   3865#   3883#   3891#   3896#   3900#   3903#   3909#
                3914#   3918#   3921#   3951#   3957#   3962#   3982#   3995#   4000#   4007#   4011#   4025#   4030#   4038#   4041#
                4060#   4073#   4078#   4085#   4089#   4103#   4108#   4116#   4119#   4138#   4151#   4156#   4163#   4167#   4181#
                4186#   4194#   4197#   4216#   4229#   4234#   4241#   4245#   4259#   4264#   4272#   4275#   4294#   4311#   4316#
                4324#   4328#   4346#   4351#   4360#   4363#   4382#   4399#   4404#   4412#   4416#   4434#   4439#   4448#   4451#
                4470#   4483#   4488#   4495#   4499#   4513#   4518#   4526#   4529#   4548#   4561#   4566#   4573#   4577#   4591#
                4596#   4604#   4607#   4626#   4639#   4644#   4651#   4655#   4669#   4674#   4682#   4685#   4704#   4717#   4722#
                4729#   4733#   4747#   4752#   4760#   4763#   4782#   4795#   4800#   4807#   4811#   4825#   4830#   4838#   4841#
                4860#   4873#   4878#   4885#   4889#   4903#   4908#   4916#   4919#   4938#   4951#   4956#   4963#   4967#   4981#
                4986#   4994#   4997#   5016#   5029#   5034#   5041#   5045#   5059#   5064#   5072#   5075#   5094#   5115#   5120#
                5128#   5132#   5143#   5148#   5157#   5160#   5179#   5192#   5197#   5204#   5208#   5223#   5228#   5238#   5241#
```

```
                5261#  5263#  5283#  5288#  5295#  5298#  5320#  5325#  5333#  5340#  5343#  5362#  5383#  5388#  5396#
                5400#  5415#  5420#  5428#  5431#  5445#  5453#  5460#  5463#  5469#  5477#  5498#  5505#  5508#  5514#
                5522#  5541#  5559#  5566#  5568#  5577#  5596#  5615#  5618#  5626#  5636#  5648#  5674#  5679#  5682#
                5688#  5726#  5731#  5734#  5740#  5776#  5781#  5784#  5790#  5847#  5854#  5901#  5911#  5950#  5965#
                5973#  6012#  6019#  6060#  6065#  6081#  6088#  6112#  6131#  6136#  6140#  6143#  6173#  6193#  6198#
                6206#  6208#  6217#  6242#  6262#  6267#  6275#  6277#  6286#  6311#  6331#  6336#  6344#  6346#  6355#
                6380#  6400#  6405#  6413#  6415#  6424#  6449#  6469#  6474#  6482#  6484#  6493#  6518#  6538#  6543#
                6551#  6553#  6562#  6587#  6607#  6612#  6620#  6622#  6631#  6656#  6676#  6681#  6689#  6691#  6700#
                6725#  6745#  6750#  6758#  6760#  6770#  6796#  6816#  6821#  6829#  6831#  6840#  6866#  6886#  6891#
                6899#  6901#  6911#  6937#  6957#  6962#  6970#  6972#  6981#  7007#  7027#  7032#  7040#  7042#  7051#
                7077#  7097#  7102#  7110#  7112#  7121#  7147#  7167#  7172#  7180#  7182#  7191#  7218#  7238#  7243#
                7251#  7253#  7262#  7288#  7308#  7313#  7321#  7323#  7332#  7359#  7379#  7384#  7392#  7394#  7403#
                7429#  7449#  7454#  7462#  7464#  7473#  7499#  7519#  7524#  7532#  7534#  7543#  7569#  7589#  7594#
                7602#  7604#  7613#  7639#  7659#  7664#  7672#  7674#  7683#  7709#  7729#  7734#  7742#  7744#  7753#
                7779#  7799#  7804#  7812#  7814#  7823#  7849#  7869#  7874#  7882#  7884#  7893#  7919#  7939#  7944#
                7952#  7954#  7963#  7990#  8010#  8015#  8023#  8025#  8034#  8060#  8080#  8085#  8093#  8095#  8104#
                8130#  8150#  8155#  8163#  8165#  8174#  8200#  8220#  8225#  8233#  8235#  8244#  8269#  8289#  8294#
                8302#  8304#  8313#  8339#  8359#  8364#  8372#  8374#  8383#
MSTSTL             1#  1368#  2184#  2497#  2504#  2507#  2514#  2521#  2524#  2531#  2538#  2541#  2548#  2556#  2559#
                2566#  2574#  2577#  2584#  2592#  2595#  2602#  2610#  2613#  2620#  2628#  2631#  2638#  2645#  2648#
                2655#  2662#  2665#  2672#  2675#  2682#  2689#  2692#  2699#  2706#  2709#  2716#  2719#  2726#  2729#
                2736#  2743#  2746#  2753#  2756#  2763#  2766#  2773#  2780#  2783#  2790#  2793#  2800#  2807#  2810#
                2817#  2824#  2827#  2834#  2842#  2845#  2852#  2859#  2862#  2869#  2877#  2880#  2887#  2894#  2897#
                2904#  2907#  2914#  2917#  2924#  2931#  2934#  2941#  2948#  2951#  2958#  2965#  2968#  2975#  2982#
                2985#  2994#  3001#  3005#  3028#  3058#  3064#  3070#  3077#  3093#  3175#  3195#  3201#  3213#  3217#
                3234#  3237#  3255#  3283#  3292#  3300#  3304#  3321#  3328#  3359#  3366#  3371#  3379#  3383#  3389#
                3394#  3402#  3405#  3422#  3430#  3435#  3442#  3447#  3453#  3456#  3477#  3485#  3490#  3494#  3496#
                3502#  3507#  3511#  3515#  3530#  3538#  3543#  3547#  3549#  3555#  3561#  3565#  3580#  3588#  3593#
                3597#  3599#  3605#  3611#  3615#  3631#  3640#  3645#  3652#  3657#  3663#  3666#  3690#  3697#  3702#
                3709#  3712#  3720#  3725#  3733#  3736#  3754#  3761#  3766#  3773#  3776#  3785#  3790#  3798#  3801#
                3819#  3826#  3831#  3841#  3843#  3849#  3854#  3862#  3865#  3883#  3891#  3896#  3900#  3903#  3909#
                3914#  3918#  3921#  3951#  3957#  3962#  3982#  3995#  4000#  4007#  4011#  4025#  4030#  4038#  4041#
                4060#  4073#  4078#  4085#  4089#  4103#  4108#  4116#  4119#  4138#  4151#  4156#  4163#  4167#  4181#
                4186#  4194#  4197#  4216#  4229#  4234#  4241#  4245#  4259#  4264#  4272#  4275#  4294#  4311#  4316#
                4324#  4328#  4346#  4351#  4360#  4363#  4382#  4399#  4404#  4412#  4416#  4434#  4439#  4448#  4451#
                4470#  4483#  4488#  4495#  4499#  4513#  4518#  4526#  4529#  4548#  4561#  4566#  4573#  4577#  4591#
                4596#  4604#  4607#  4626#  4639#  4644#  4651#  4655#  4669#  4674#  4682#  4685#  4704#  4717#  4722#
                4729#  4733#  4747#  4752#  4760#  4763#  4782#  4795#  4800#  4807#  4811#  4825#  4830#  4838#  4841#
                4860#  4873#  4878#  4885#  4889#  4903#  4908#  4916#  4919#  4938#  4951#  4956#  4963#  4967#  4981#
                4986#  4994#  4997#  5016#  5029#  5034#  5041#  5045#  5059#  5064#  5072#  5075#  5094#  5115#  5120#
                5128#  5132#  5143#  5148#  5157#  5160#  5179#  5192#  5197#  5204#  5208#  5223#  5228#  5238#  5241#
```

	7952#	7954#	7563#	7990#	8010#	8015#	8023#	8025#	8034#	8060#	8080#	8085#	8093#	8095#	8104#
	8130#	8150#	8155#	8163#	8165#	8174#	8200#	8220#	8225#	8233#	8235#	8244#	8269#	8289#	8294#
	8302#	8304#	8313#	8339#	8359#	8364#	8372#	8374#	8383#						
MSWORD	1#	1368#	1440#	1449	1504#	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515
	1516	1517	1518	1519	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530
	1531	1532	1533	1534	1535	1536	1537	1538	1539	1540	1541	1542	1543	1544	1545
	1546	1547	1548	1549	1550	1551	1552	1553	1554	1555	1556	1557	1558	1559	1560
	1561	1562	1563	1564	1565	1566	1567	1568	1569	1570	1571	1572	1573	1574	1575
	1576	1577	1578	1579	1580	1581	2184#	2185	2186	2187	3022#	3292#	3293	3294	3295
	3321#	3322	3323	3324	3366#	3367	3368	3369	3389#	3390	3391	3392	3430#	3431	3432
	3433	3442#	3443	3444	3445	3485#	3486	3487	3488	3502#	3503	3504	3505	3538#	3539
	3540	3541	3555#	3556	3557	3558	3588#	3589	3590	3591	3605#	3606	3607	3608	3640#
	3641	3642	3643	3652#	3653	3654	3655	3697#	3698	3699	3700	3720#	3721	3722	3723
	3761#	3762	3763	3764	3785#	3786	3787	3788	3826#	3827	3828	3829	3849#	3850	3851
	3852	3891#	3892	3893	3894	3909#	3910	3911	3912	3951#	3952	3953	3954	3995#	3996
	3997	3998	4025#	4026	4027	4028	4073#	4074	4075	4076	4103#	4104	4105	4106	4151#
	4152	4153	4154	4181#	4182	4183	4184	4229#	4230	4231	4232	4259#	4260	4261	4262
	4311#	4312	4313	4314	4346#	4347	4348	4349	4399#	4400	4401	4402	4434#	4435	4436
	4437	4483#	4484	4485	4486	4513#	4514	4515	4516	4561#	4562	4563	4564	4591#	4592
	4593	4594	4639#	4640	4641	4642	4669#	4670	4671	4672	4717#	4718	4719	4720	4747#
	4748	4749	4750	4795#	4796	4797	4798	4825#	4826	4827	4828	4873#	4874	4875	4876
	4903#	4904	4905	4906	4951#	4952	4953	4954	4981#	4982	4983	4984	5029#	5030	5031
	5032	5059#	5060	5061	5062	5115#	5116	5117	5118	5143#	5144	5145	5146	5192#	5193
	5194	5195	5223#	5224	5225	5226	5283#	5284	5285	5286	5320#	5321	5322	5323	5383#
	5384	5385	5386	5415#	5416	5417	5418	5463#	5464	5465	5466	5469#	5470	5471	5472
	5508#	5509	5510	5511	5514#	5515	5516	5517	5568#	5569	5570	5571	5618#	5619	5620
	5621	5626#	5627	5628	5629	5674#	5675	5676	5677	5682#	5726#	5727	5728	5729	5734#
	5776#	5777	5778	5779	5784#	5847#	5848	5849	5850	5901#	5902	5903	5904	5950#	5951
	5952	5953	5965#	5966	5967	5968	6012#	6013	6014	6015	6060#	6061	6062	6063	6081#
	6131#	6132	6133	6134	6193#	6194	6195	6196	6208#	6252#	6263	6264	6265	6277#	6331#
	6332	6333	6334	6346#	6400#	6401	6402	6403	6415#	6469#	6470	6471	6472	6484#	6538#
	6539	6540	6541	6553#	6607#	6608	6609	6610	6622#	6676#	6677	6678	6679	6691#	6745#
	6746	6747	6748	6760#	6816#	6817	6818	6819	6831#	6886#	6887	6888	6889	6901#	6957#
	6958	6959	6960	6972#	7027#	7028	7029	7030	7042#	7097#	7098	7099	7100	7112#	7167#
	7168	7169	7170	7182#	7238#	7239	7240	7241	7253#	7308#	7309	7310	7311	7323#	7379#
	7380	7381	7382	7394#	7449#	7450	7451	7452	7464#	7519#	7520	7521	7522	7534#	7589#
	7590	7591	7592	7604#	7659#	7660	7661	7662	7674#	7729#	7730	7731	7732	7744#	7799#
	7800	7801	7802	7814#	7869#	7870	7871	7872	7884#	7939#	7940	7941	7942	7954#	8010#
	8011	8012	8013	8025#	8080#	8081	8082	8083	8095#	8150#	8151	8152	8153	8165#	8220#
	8221	8222	8223	8235#	8289#	8290	8291	8292	8304#	8359#	8360	8361	8362	8374#	8410#
	8416#	8421#	8426#	8542	8543										
MSXFER	1#	1368#													
OPEN	1#	1368#													
POINTE	1#	1368#	1397												
PRINTB	1#	1368#	2493	2499	2510	2516	2527	2533	2544	2550	2562	2568	2580	2586	2598
	2604	2616	2622	2634	2640	2651	2657	2668	2678	2684	2695	2701	2712	2722	2732
	2738	2749	2759	2769	2775	2786	2796	2802	2813	2819	2830	2836	2848	2854	2865
	2871	2883	2889	2900	2910	2920	2926	2937	2943	2954	2960	2971	2977		
PRINTF	1#	1368#	2989	2996											
PRINTS	1#	1368#													
PRINTX	1#	1368#													
READBU	1#	1368#													
READEF	1#	1368#	3056	3062	3068	3075									
RERROR	1939#	5282	5319	5382	5414										
RFLAGS	1#	1368#													
ROMCLK	1978#	2014	2017	2020	2027	2035	2043	2051	2060	2075	2138	2157	2160	2164	2170

```
            2175    2205    2217    2220    2229    2232    3985    3988    4015    4018    4063    4066    4093    4096    4141
            4144    4171    4174    4219    4222    4249    4252    4298    4301    4333    4336    4386    4389    4421    4424
            4473    4476    4503    4506    4551    4554    4581    4584    4629    4632    4659    4662    4707    4710    4737
            4740    4785    4788    4815    4818    4863    4866    4893    4896    4941    4944    4971    4974    5019    5022
            5049    5052    5099    5108    5136    5182    5185    5212    5215    5267    5272    5275    5304    5309    5312
            5367    5372    5375    5404    5407    5455    5500    5554    5610    5661    5665    5711    5716    5719    5763
            5767    5814    5837    5840    5889    5892    5933    5937    5941    5957    5995    5999    6003    6051    6114
            6117    6120    6123    6177    6182    6185    6246    6251    6254    6315    6320    6323    6384    6389    6392
            6453    6458    6461    6522    6527    6530    6591    6596    6599    6660    6665    6668    6729    6734    6737
            6800    6805    6808    6870    6875    6878    6941    6946    6949    7011    7016    7019    7081    7086    7089
            7151    7156    7159    7222    7227    7230    7292    7297    7300    7363    7368    7371    7433    7438    7441
            7503    7508    7511    7573    7578    7581    7643    7648    7651    7713    7718    7721    7783    7788    7791
            7853    7858    7861    7923    7928    7931    7994    7999    8002    8064    8069    8072    8134    8139    8142
            8204    8209    8212    8273    8278    8281    8343    8348    8351
SETPRI         1#  1368#   5451    5458    5496    5503    5539    5557    5594    5613
SETVEC         1#  1368#
SLASH          1#  1368#
STARS          1#  1368#
SVC            1#  1366#   1367
XFER           1#  1368#   3022#   5682#   5734#   5784#   6081#   6208#   6277#   6346#   6415#   6484#   6553#   6622#   6691#
            6760#   6831#   6901#   6972#   7042#   7112#   7182#   7253#   7323#   7394#   7464#   7534#   7604#   7674#   7744#
            7814#   7884#   7954#   8025#   8095#   8165#   8235#   8304#   8374#
XFERF          1#  1368#
XFERT          1#  1368#
$MD         2476#   2491    2508    2525    2542    2560    2578    2596    2614    2632    2649    2666    2676    2693    2710
            2720    2730    2747    2757    2767    2784    2794    2811    2828    2846    2863    2881    2898    2908    2918
            2935    2952    2969

. ABS.  040004     000

 ERRORS DETECTED:  0

 CZDMPD,CZDMPD/SOL/CRF/NL:TOC=SVC34R.MLB,CZDMPD.P11
 RUN-TIME: 47 56 7 SECONDS
 RUN-TIME RATIO: 198/111=1.7
 CORE USED:  24K  (47 PAGES)
```