DMP-11,DMR-11, M8207 STATIC DIAG#1 AH-E226C-MC OCT 1981
M8207 CZDMPC0 FICHE 1 OF 2

DMP-11,DMR-11, M8207 STATIC DIAG#1 AH-E226C-MC OCT 1981
M8207 CZDMPC0 FICHE 2 OF 2

```
2197                                    .REM @
2198
2199                                                IDENTIFICATION
2200                                                --------------
2201
2202                                    PRODUCT CODE:   AC-E225C-MC
2203
2204                                    PRODUCT NAME:   CZDMPCO M8207 STATIC DIAG #1
2205
2206                                    PRODUCT DATE:   JULY 1981
2207
2208                                    MAINTAINER:     DIAGNOSTICS MERRIMACK
2209
2210                                    AUTHOR:         ED BADGER
2211
2212
2213
2214
2215
2216                                        THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
2217                                        NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
2218                                        EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO
2219                                        RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.
2220
2221                                        NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF
2222                                        SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS
2223                                        AFFILIATED COMPANIES.
2224
2225                                        COPYRIGHT (C) 1979,1981 BY DIGITAL EQUIPMENT CORPORATION
2226
2227                                    THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:
2228
2229                                            DIGITAL         PDP             UNIBUS          MASSBUS
2230                                            DEC             DECUS           DECTAPE
2231
```

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 PROGRAM ABSTRACT

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

THE PROGRAM WAS IMPLEMENTED USING THE DIAGNOSTIC SUPERVISOR.

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

1.2 HARDWARE INTRODUCTION

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

2.0 HARDWARE REQUIREMENTS

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

PDP-11/04,05,10,20,30,34,35,40,45,50,60, OR 70
16K MEMORY
CONSOLE TERMINAL

3.0 PRELIMINARY PROGRAM REQUIREMENTS

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

4.0 GENERAL PROGRAM CONSIDERATIONS

4.1 DIAGNOSTIC SUPERVISOR

THIS PROGRAM IS COMPATIBLE WITH THE STANDALONE DIAGNOSTIC

2341 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-C-0
```

2455 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) CONTAIN-
                              ING THE ERROR
                        IER   INHIBIT ERROR REPORTING
                        IBE   INHIBIT BASIC ERROR REPORTS
                        IXE   INHIBIT EXTENDED ERROR REPORTS
                        PRI   DIRECT ALL MESSAGES TO A LINE PRINTER
                        PNT   PRINT NUMBER OF TEST BEING EXECUTED
                        BOE   BELL ON ERROR
                        UAM   RUN IN UNATTENDED MODE, BYPASSING MANUAL
                              INTERVENTION TESTS
                        ISR   INHIBIT STATISTICAL REPORTS
                        IDU   INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
                        LOT   LOOP ON TEST

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


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

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


                    6.3.1.5 EFFECT OF START COMMAND

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

                    THE  HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION
                    "# UNITS?" TO WHICH THE  OPERATOR  REPLIES  WITH  A  DECIMAL
                    NUMBER  N FROM 1 TO 16. THE TERM 'UNIT" REFERS TO THE DEVICE
                    TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED.  FOLLOWING
                    THIS  ARE THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES WILL
                    BE BUILT.  EACH P-TABLE IS A CORE-RESIDENT TABLE  CONTAINING
                    ALL  THE  HARDWARE  INFORMATION  FOR ONE UNIT.  THE OPERATOR
                    MUST SUPPLY N (NUMBER OF UNITS) VALUES  FOR  EACH  QUESTION.
                    HE  MAY  DO  THIS  BY GIVING ONE ANSWER TO EACH QUESTION (IN
                    WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR
                    BY  GIVING  N  VALUES, SEPARATED BY COMMAS, TO EACH QUESTION
                    (SERIES WILL BE POSED ONCE).  EACH QUESTION IS  FOLLOWED  BY
                    THE  RESPONSE  RADIX  (D  FOR  DECIMAL,  B FOR BINARY, O FOR
```

```
2570                                    OCTAL, L FOR YES/NO) IN PARENTHESES AND  THE  DEFAULT  VALUE
```

AFTER THE PARENTHESES.

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

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

EXAMPLE:

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

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

6.3.2 RESTART COMMAND

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

6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

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

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

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

2628 COMMAND.

6.3.2.3 EFFECT OF RESTART COMMAND

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

6.3.3 CONTINUE COMMAND

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

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

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

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

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

6.3.3.3 EFFECT OF CONTINUE COMMAND

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

6.3.4 PROCEED COMMAND

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

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

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

6.3.4.2 EFFECT OF PROCEED COMMAND

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

6.3.5 ADD COMMAND

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

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

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

6.3.5.2 EFFECT OF ADD COMMAND

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

6.3.6 DROP COMMAND

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

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

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

6.3.6.2 EFFECT OF DROP COMMAND

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

6.3.7 PRINT COMMAND

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

6.3.7.1 EFFECT OF PRINT COMMAND

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

6.3.8 DISPLAY COMMAND

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

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

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

6.3.8.2 EFFECT OF DISPLAY COMMAND

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

6.3.9 FLAGS COMMAND

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

6.3.9.1 EFFECT OF FLAGS COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

6.3.10 ZFLAGS COMMAND

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

6.3.10.1 EFFECT OF ZFLAGS COMMAND

ALL FLAGS ARE CLEARED.

6.3.11 CONTROL CHARACTERS

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

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

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

6.3.12 HARDWARE PARAMETERS

THE FOLLOWING 4 QUESTION WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIGE RETURN RESPONSE.

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

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

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

THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SEL0) RESIDE ON THE UNIBUS. THE ALLOWABLE RANGE IS 60000-177776 (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).

```
2965
2966
2967
2968
2969
2970                            7.0 TEST DESCRIPTION
2971
2972
2973
2974                            ******************** TEST 1 ********************
2975                            *VERIFY THAT REFERENCING UNIBUS DEVICE REGISTERS
2976                            *DOES NOT CAUSE A TIMEOUT TRAP
2977                            ************************************************
2978
2979
2980                            ******************** TEST 2 ********************
2981                            *VERIFY THAT RUN CAN BE CLEARED
2982                            ************************************************
2983
2984                            ******************** TEST 3 ********************
2985                            *UNIBUS REGISTER WORD DUAL ADDRESSING TEST
2986                            *LOAD ALL REGISTERS WITH INCREMENTING PATTERN
2987                            *READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING
2988                            *THE SEQUENCE:
2989                            *       1.  CLEAR REGISTER
2990                            *       2.  WRITE PATTERN
2991                            *       3   VERIFY PATTERN
2992                            *       ..  DO ALL 4 REGISTERS
2993                            *       5.  READ ALL BACK IF ERRORS,
2994                            *           DUAL ADDRESS PROBLEM.
2995                            *
2996                            *           1 IN REG 0
2997                            *           2 IN REG 2
2998                            *           3 IN REG 4
2999                            *           4 IN REG 6
3000                            ************************************************
3001
3002
3003                            ******************** TEST 4 ********************
3004                            *CONTROL STATUS REGISTER WRITE/READ TEST
3005                            *FLOAT A ONE THROUGH BSEL 0
3006                            *CLEAR BIT0, VERIFY BIT0 WAS CLEARED
3007                            ************************************************
3008
3009
3010                            ******************** TEST 5 ********************
3011                            *CONTROL STATUS REGISTER WRITE/READ TEST
3012                            *SET BIT9, VERIFY BIT9 WAS SET
3013                            *CLEAR BIT9, VERIFY BIT9 WAS CLEARED
3014                            ************************************************
3015
3016
3017                            ******************** TEST 6 ********************
3018                            *CONTROL STATUS REGISTER WRITE/READ TEST
3019                            *SET BIT11, VERIFY BIT11 WAS SET
3020                            *CLEAR BIT11, VERIFY BIT11 WAS CLEARED
```

```
3021                                    **************************************************
```

```
3023
3024
3025
3026
3027
3028
3029
3030                    ;******************** TEST 7 ********************
3031                    ;CONTROL STATUS REGISTER WRITE/READ TEST
3032                    ;SET BIT12, VERIFY BIT12 WAS SET
3033                    ;CLEAR BIT 12, VERIFY BIT 12 WAS CLEARED
3034                    ;************************************************
3035
3036
3037                    ;******************** TEST 8 ********************
3038                    ;CONTROL OUT REGISTER WRITE/READ TEST
3039                    ;FLOAT A ONE THROUGH SEL2
3040                    ;************************************************
3041
3042
3043                    ;******************** TEST 9 ********************
3044                    ;PORT4 REGISTER WRITE/READ TEST
3045                    ;FLOAT A ONE THROUGH PORT4 REGISTER
3046                    ;FLOAT A ZERO THROUGH PORT4 REGISTER
3047                    ;************************************************
3048
3049
3050                    ;******************** TEST 10 ********************
3051                    ;PORT6 REGISTER WRITE/READ TEST
3052                    ;FLOAT A ONE THROUGH PORT6 REGISTER
3053                    ;FLOAT A ZERO THROUGH PORT6 REGISTER
3054                    ;************************************************
3055
3056
3057                    ;******************** TEST 11 ********************
3058                    ;UNIBUS REGISTER BYTE DUAL ADDRESSING TEST
3059                    ;LOAD ALL REGISTERS WITH INCREMENTING PATTERN
3060                    ;READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING
3061                    ;************************************************
3062
3063
3064                    ;******************** TEST 12 ********************
3065                    ;MAINTENANCE INSTRUCTION REGISTER TEST
3066                    ;VERIFY THAT THE MAINT IR CAN BE WRITTEN TO ALL ZEROS'
3067                    ;AND ALL ONES'.  VERIFY THAT IS IS CLEARED ON A BUS RESET.
3068                    ;************************************************
3069
3070
3071                    ;******************** TEST 13 ********************
3072                    ;MICRO PROCESSOR TEST
3073                    ;LOAD KMP06 WITH A MICRO-PROCESSOR INSTRUCTION, CLOCK IT
3074                    ;VERIFY INSTRUCTION EXECUTED PROPERLY
3075                    ;INSTRUCTION SHOULD MOVE IBUS*4 TO IBUS*5, IBUS*4 IS ALL 1'S
3076                    ;AND IBUS*5 IS ALL 0'S. RESULT SHOULD BE ALL 1'S IN SEL4
3077                    ;************************************************
3078
```

3079

```
3081
3082
3083
3084
3085
3086                    ********** TEST 14 **********
3087                    *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
3088                    *FLOAT A 1 THROUGH IBUS* REGISTER 0
3089                    *FLOAT A 0 THROUGH IBUS* REGISTER 0
3090                    ************************************************
3091
3092
3093                    ********** TEST 15 **********
3094                    *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
3095                    *FLOAT A 1 THROUGH IBUS* REGISTER 2
3096                    *FLOAT A 0 THROUGH IBUS* REGISTER 2
3097                    ************************************************
3098
3099
3100                    ********** TEST 16 **********
3101                    *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
3102                    *FLOAT A 1 THROUGH IBUS* REGISTER 4
3103                    *FLOAT A 0 THROUGH IBUS* REGISTER 4
3104                    ************************************************
3105
3106
3107                    ********** TEST 17 **********
3108                    *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
3109                    *FLOAT A 1 THROUGH IBUS* REGISTER 5
3110                    *FLOAT A 0 THROUGH IBUS* REGISTER 5
3111                    ************************************************
3112
3113
3114                    ********** TEST 18 **********
3115                    *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
3116                    *FLOAT A 1 THROUGH IBUS* REGISTER 10
3117                    *FLOAT A 0 THROUGH IBUS* REGISTER 10
3118                    ************************************************
3119
3120
3121                    ********** TEST 19 **********
3122                    *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
3123                    *FLOAT A 1 THROUGH IBUS* REGISTER 11
3124                    *FLOAT A 0 THROUGH IBUS* REGISTER 11
3125                    ************************************************
3126
3127
3128                    ********** TEST 20 **********
3129                    *MICRO PROCESSOR IBUS REBISTER WRITE/READ TEST
3130                    *FLOAT A 1 THROUGH IBUS REGISTER 0
3131                    *FLOAT A 0 THROUGH IBUS REGISTER 0
3132                    ************************************************
3133
3134
3135                    ********** TEST 21 **********
3136                    *MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
```

```
3137                                    *FLOAT A 1 THROUGH IBUS REGISTER 1
```

```
3139
3140
3141
3142
3143
3144                    *FLOAT A 0 THROUGH IBUS REGISTER 1
3145                    ****************************************** ********
3146
3147
3148                    ********************** TEST 22 *****************
3149                    *MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
3150                    *FLOAT A 1 THROUGH IBUS REGISTER 2
3151                    *FLOAT A 0 THROUGH IBUS REGISTER 2
3152                    ************************************************
3153
3154
3155                    ********************** TEST 23 *****************
3156                    *MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
3157                    *FLOAT A 1 THROUGH IBUS REGISTER 3
3158                    *FLOAT A 0 THROUGH IBUS REGISTER 3
3159                    ************************************************
3160
3161
3162                    ********************** TEST 24 *****************
3163                    *MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
3164                    *FLOAT A 1 THROUGH IBUS REGISTER 4
3165                    *FLOAT A 0 THROUGH IBUS REGISTER 4
3166                    ************************************************
3167
3168
3169                    ********************** TEST 25 *****************
3170                    *MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
3171                    *FLOAT A 1 THROUGH IBUS REGISTER 5
3172                    *FLOAT A 0 THROUGH IBUS REGISTER 5
3173                    ************************************************
3174
3175
3176                    ********************** TEST 26 *****************
3177                    *MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
3178                    *FLOAT A 1 THROUGH IBUS REGISTER 6
3179                    *FLOAT A 0 THROUGH IBUS REGISTER 6
3180                    ************************************************
3181
3182
3183                    ********************** TEST 27 *****************
3184                    *MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
3185                    *FLOAT A 1 THROUGH IBUS* REGISTER 7
3186                    *FLOAT A 0 THROUGH IBUS* REGISTER 7
3187                    ************************************************
3188
3189
3190                    ********************** TEST 28 *****************
3191                    *MICRO PROCESSOR IBUS DUAL ADDRESS TEST
3192                    *WRITE ALL IBUS REGISTERS WITH INCREMENTING PATTERN
3193                    *READ ALL IBUS REGISTERS TO VERIFY CORRECT ADDRESSING
3194                    ************************************************
```

3195

```
3197
3198
3199
3200
3201
3202
3203                    ********** TEST 29 ******** *********
3204                    *MICRO PROCESSOR BR REGISTER TEST
3205                    *FLOAT A 1 THROUGH THE BR
3206                    *FLOAT A 0 THROUGH THE BR
3207                    ************************************************
3208
3209
3210                    ********** TEST 30 *****************
3211                    *SCRATCH PAD TEST
3212                    *FLOAT A 1 THROUGH EACH SCRATCH PAD LOCATION
3213                    *FLOAT A 0 THROUGH EACH SCRATCH PAD LOCATION
3214                    ************************************************
3215
3216
3217                    ********** TEST 31 *****************
3218                    *SCRATCH PAD DUAL ADDRESSING TEST
3219                    *WRITE AN INCREMENTING PATTERN IN ALL S  '  TIONS
3220                    *READ ALL SP LOCATIONS TO VERIFY CORREC  RESSING
3221                    ************************************************
3222
3223
3224                    ********** TEST 32 *****************
3225                    *INTERRUPT TEST
3226                    *TEST THAT DEVICE CAN INTERRUPT TO VECTOR A
3227                    ************************************************
3228
3229
3230                    ********** TEST 33 *****************
3231                    *INTERRUPT TEST
3232                    *TEST THAT DEVICE CAN INTERRUPT TO VECTOR B
3233                    ************************************************
3234
3235
3236                    ********** TEST 34 *****************
3237                    *PRIORITY INTERRUPT TEST
3238                    *SET PS TO ALL BR LEVELS EQUAL OR GREATER THAN
3239                    *THE M8200,4,7 LEVEL, VERIFY THAT M8200,4,7 DOES NOT INTERRUPT
3240                    ************************************************
3241
3242
3243                    ********** TEST 35 *****************
3244                    *PRIORITY INTERRUPT TESTS
3245                    *SET PS TO ALL BR LEVELS LESS THAN THE M8200,4,7 LEVEL
3246                    *VERIFY THAT ALL M8200,4,7 WILL INTERRUPT
3247                    ************************************************
3248
3249
3250                    ********** TEST 36 *****************
3251                    *NPR TEST
3252                    *TEST OF DATO, 1 WORD FROM UPROC TO 11 MEMORY
```

```
3253                ;**********************************************
```

```
3255
3256
3257
3258
3259
3260
3261
3262                    ********** TEST 37 **********
3263                    *NPR TEST
3264                    *TEST OF DATI, 1 WORD FROM 11 MEMORY TO UPROC
3265                    ************************************************
3266
3267
3268                    ********** TEST 38 **********
3269                    *NPR TEST
3270                    *TEST OF DATOB, 1 BYTE FROM UPROC TO 11 MEMORY
3271                    ************************************************
3272
3273
3274                    ********** TEST 39 **********
3275                    *TEST OF EA BITS 16 AND 17
3276                    *DO A DATO TO AN ADDRESS USING OUT BA BITS 16 AND 17
3277                    *VERIFY CORRECT RESULTS
3278                    ************************************************
3279
3280
3281                    ********** TEST 40 **********
3282                    *TEST OF EA BITS 16 AND 17
3283                    *DO A DATI USING IN BA BITS 16 AND 17
3284                    *VERIFY CORRECT RESULTS
3285                    *IN ORDER TO DO THIS TEST, WE WILL READ THE DATA FROM THE
3286                    *CONSOL TTY CSR IF ONE EXISTS
3287                    *IF NO COSOL TTY CSR AT ADDRESS 177560, THIS TEST
3288                    *WILL BE SKIPPED
3289                    ************************************************
3290
3291
3292                    ********** TEST 41 **********
3293                    *NPR NON-EXISTENT MEMORY TEST
3294                    *DO A DATO TO A NON -EXISTENT ADDRESS
3295                    *VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
3296                    ************************************************
3297
3298
3299                    ********** TEST 42 **********
3300                    *NPR NON-EXISTENT MEMORY TEST
3301                    *DO A DATI FROM A NON-EXISTENT ADDRESS
3302                    *VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
3303                    ************************************************
3304
3305
3306                    ********** TEST 43 **********
3307                    *NPR TEST
3308                    *USING DATO, NPR A BINARY COUNT (0-377)
3309                    *FROM MICRO-PROCESSOR TO ALL AVAILABLE MEMORY
3310                    ************************************************
```

E 3

3311

```
3313
3314
3315
3316
3317
3318
3319                    ****************** TEST 44 ********* *********
3320                    *ALU C BIT TEST
3321                    *TEST THAT AN ADD OF 377 AND 377 WILL SET THE C BIT
3322                    ***************************************************
3323
3324
3325                    ****************** TEST 45 ********* *********
3326                    *ALU TEST
3327                    *TEST OF ALU FUNCTION SEL B WITH C BIT CLEARED
3328                    *ALU FUNCTION (B)    CODE=11
3329                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3330                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3331                    ***************************************************
3332
3333
3334                    ****************** TEST 46 ********* *********
3335                    *ALU TEST
3336                    *TEST OF ALU FUNCTION SEL A WITH C BIT CLEARED
3337                    *ALU FUNCTION (A)    CODE-10
3338                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3339                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3340                    ***************************************************
3341
3342
3343                    ****************** TEST 47 ********* *********
3344                    *ALU TEST
3345                    *TEST OF ALU FUNCTION A OR NOTB WITH C BIT CLEARED
3346                    *ALU FUNCTION (A OR NOTB)    CODE=12
3347                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3348                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3349                    ***************************************************
3350
3351
3352                    ****************** TEST 48 ********* *********
3353                    *ALU TEST
3354                    *TEST OF ALU FUNCTION A AND B  WITH C BIT CLEARED
3355                    *ALU FUNCTION (A AND B)    CODE=13
3356                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3357                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3358                    ***************************************************
3359
3360
3361                    ****************** TEST 49 ********* *********
3362                    *ALU TEST
3363                    *TEST OF ALU FUNCTION A OR B WITH C BIT CLEARED
3364                    *ALU FUNCTION  (A OR B)    CODE-14
3365                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3366                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3367                    ***************************************************
3368
```

3369

```
3371
3372
3373
3374
3375
3376                    ********** TEST 50 **********
3377                    *ALU TEST
3378                    *TEST OF ALU FUNCTION A XOR B WITH C BIT
3379                    *ALU FUNCTION (A XOR B)    CODE=15
3380                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3381                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3382                    ************************************************
3383
3384
3385                    ********** TEST 51 **********
3386                    *ALU TEST
3387                    *TEST OF ALU FUNCTION ADD WITH C BIT CLEARED
3388                    *ALU FUNCTION (A PLUS B)    CODE=00
3389                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3390                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3391                    ************************************************
3392
3393
3394                    ********** TEST 52 **********
3395                    *ALU TEST
3396                    *TEST OF ALU FUNCTION 2A W.C WITH C BIT CLEARED
3397                    *ALU FUNCTION (A PLUS A PLUS C)    CODE-6
3398                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3399                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3400                    ************************************************
3401
3402
3403                    ********** TEST 53 **********
3404                    *ALU TEST
3405                    *TEST OF ALU FUNCTION SUB WITH C BIT CLEARED
3406                    *ALU FUNCTION (A-B)    CODE=16
3407                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3408                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3409                    ************************************************
3410
3411
3412                    ********** TEST 54 **********
3413                    *ALU TEST
3414                    *TEST OF ALU FUNCTION ADD W/C WITH C BIT CLEARED
3415                    *ALU FUNCTION (A PLUS B PLUS C)    CODE=01
3416                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3417                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3418                    ************************************************
3419
3420
3421                    ********** TEST 55 **********
3422                    *ALU TEST
3423                    *TEST OF ALU FUNCTION SUB W/C WITH C BIT CLEARED
3424                    *ALU FUNCTION (A-B-C)    CODE-2
3425                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3426                    *PERFORM THE FUNCTION. VERIFY THE RESULTS
```

```
3427                                  ************************************************
```

```
3429
3430
3431
3432
3433
3434
3435
3436                    ********** TEST 56 **********
3437                    *ALU TEST
3438                    *TEST OF ALU FUNCTION INC A WITH C BIT CLEARED
3439                    *ALU FUNCTION (A PLUS 1)    CODE =3
3440                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3441                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3442                    ************************************************
3443
3444
3445                    ********** TEST 57 **********
3446                    *ALU TEST
3447                    *TEST OF ALU FUNCTION 2A WITH C BIT CLEARED
3448                    *ALU FUNCTION (A PLUS A)    CODE=5
3449                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3450                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3451                    ************************************************
3452
3453
3454                    ********** TEST 58 **********
3455                    *ALU TEST
3456                    *TEST OF ALU FUNCTION A PLUS C WITH C BIT CLEARED
3457                    *ALU FUNCTION (A PLUS C)    CODE=4
3458                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3459                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3460                    ************************************************
3461
3462
3463                    ********** TEST 59 **********
3464                    *ALU TEST
3465                    *TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT CLEARED
3466                    *ALU FUNCTION (A-B-1)    CODE=17
3467                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3468                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3469                    ************************************************
3470
3471
3472                    ********** TEST 60 **********
3473                    *ALU TEST
3474                    *TEST OF ALU FUNCTION DEC A WITH C BIT CLEARED
3475                    *ALU FUNCTION (A-1)    CODE=7
3476                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3477                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3478                    ************************************************
3479
3480
3481                    ********** TEST 61 **********
3482                    *ALU TEST
3483                    *TEST OF ALU FUNCTION SEL B WITH C BIT SET
3484                    *ALU FUNCTION (B)    CODE=11
```

```
3485                            ;LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
```

```
3487
3488
3489
3490
3491
3492                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3493                    ********************************************* **********
3494
3495
3496                    ******************** TEST 62 **********************
3497                    *ALU TEST
3498                    *TEST OF ALU FUNCTION SEL A WITH C BIT SET
3499                    *ALU FUNCTION (A)    CODE-10
3500                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3501                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3502                    ****************************************************
3503
3504
3505                    ******************** TEST 63 **********************
3506                    *ALU TEST
3507                    *TEST OF ALU FUNCTION A OR NOTB WITH C BIT SET
3508                    *ALU FUNCTION (A OR NOTB)    CODE=12
3509                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3510                    *PERFORM THE FUNCTION, VERIFY THE RESJLTS
3511                    ****************************************************
3512
3513
3514                    ******************** TEST 64 **********************
3515                    *ALU TEST
3516                    *TEST OF ALU FUNCTION A AND B WITH C BIT SET
3517                    *ALU FUNCTION (A AND B)    CODE=13
3518                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3519                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3520                    ****************************************************
3521
3522
3523                    ******************** TEST 65 **********************
3524                    *ALU TEST
3525                    *TEST OF ALU FUNCTION A OR B WITH C BIT SET
3526                    *ALU FUNCTION (A OR B)    CODE=14
3527                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3528                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3529                    ****************************************************
3530
3531
3532                    ******************** TEST 66 **********************
3533                    *ALU TEST
3534                    *TEST OF ALU FUNCTION A XOR B WITH C BIT SET
3535                    *ALU FUNCTION (A XOR B)    CODE=15
3536                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3537                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3538                    ****************************************************
3539
3540
3541                    ******************** TEST 67 **********************
3542                    *ALU TEST
```

```
 3543                                   *TEST OF ALU FUNCTION ADD WITH C BIT SET
```

```
3545
3546
3547
3548
3549
3550                    *ALU FUNCTION (A PLUS B)      CODE=00
3551                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3552                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3553                    ****************************************************
3554
3555
3556                    ******************** TEST 68 **********************
3557                    *ALU TEST
3558                    *TEST OF ALU FUNCTION 2A W/C WITH C BIT SET
3559                    *ALU FUNCTION (A PLUS A PLUS C)     CODE=6
3560                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3561                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3562                    ****************************************************
3563
3564
3565                    ******************** TEST 69 **********************
3566                    *ALU TEST
3567                    *TEST OF ALU FUNCTION SUB WITH C BIT SET
3568                    *ALU FUNCTION (A-B)     CODE=16
3569                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3570                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3571                    ****************************************************
3572
3573
3574                    ******************** TEST 70 **********************
3575                    *ALU TEST
3576                    *TEST OF ALU FUNCTION ADD W/C WITH C BIT SET
3577                    *ALU FUNCTION (A PLUS B PLUS C)     CODE=01
3578                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3579                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3580                    ****************************************************
3581
3582
3583                    ******************** TEST 71 **********************
3584                    *ALU TEST
3585                    *TEST OF ALU FUNCTION SUB W/C WITH C BIT SET
3586                    *ALU FUNCTION (A-B-C)     CODE=2
3587                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3588                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3589                    ****************************************************
3590
3591
3592                    ******************** TEST 72 **********************
3593                    *ALU TEST
3594                    *TEST OF ALU FUNCTION INC A WITH C BIT SET
3595                    *ALU FUNCTION (A PLUS 1)     CODE=3
3596                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3597                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3598                    ****************************************************
3599
3600
```

3601 ********** TEST 73 **********

```
3603
3604
3605
3606
3607
3608                    *ALU TEST
3609                    *TEST OF ALU FUNCTION 2A WITH C BIT SET
3610                    *ALU FUNCTION (A PLUS A)    CODE=5
3611                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3612                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3613                    ************************************************
3614
3615
3616                    ******************** TEST 74 *********************
3617                    *ALU TEST
3618                    *TEST OF ALU FUNCTION A PLUS C WITH C BIT SET
3619                    *ALU FUNCTION (A PLUS C)    CODE=4
3620                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3621                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3622                    ************************************************
3623
3624
3625                    ******************** TEST 75 *********************
3626                    *ALU TEST
3627                    *TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT SET
3628                    *ALU FUNCTION (A-B-1)    CODE=17
3629                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3630                    *PERFORM THE FUNCTION, VERIFY THE RESULTS
3631                    ************************************************
3632
3633
3634                    ******************** TEST 76 *********************
3635                    *ALU TEST
3636                    *TEST OF ALU FUNCTION DEC A WITH C BIT SET
3637                    *ALU FUNCTION (A-1)    CODE=7
3638                    *LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
3639                    *PERFORM THE FUNCTION, VERIFY THE RESULT
3640                    ************************************************
3641
3642
3643
```

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.

@

```
3684
3685
3686
3687
3688
```

```
                                .TITLE CZDMPC0 M8207 STATIC DIAG #1
002000                          .=2000


                                .MCALL  SVC
002000                          SVC                             ; INITIALIZE SUPERVISOR MACROS


002000                          BGNMOD  CZDMP

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

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


                        .ENABL  AMA
```

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

                                                POINTER BGNAU,BGNDU


                                                HEADER  CZDMP,C,0,120.,0
(4)  002000                             L$NAME::                ;DIAGNOSTIC NAME
(4)  002000      103                            .ASCII /C/
(4)  002001      132                            .ASCII /Z/
(4)  002002      104                            .ASCII /D/
(4)  002003      115                            .ASCII /M/
(4)  002004      120                            .ASCII /P/
(6)  002005      000                            .BYTE   0
(6)  002006      000                            .BYTE   0
(5)  002007      000                            .BYTE   0
(5)  002010                             L$REV::                 ;REVISION LEVEL
(4)  002010      103                            .ASCII  /C/
(5)  002011                             L$DEPO::                ;0
(4)  002011      060                            .ASCII  /0/
(5)  002012                             L$UNIT::                ;NUMBER OF UNITS
(4)  002012  000000                             .WORD   0
(5)  002014                             L$TIML::                ;LONGEST TEST TIME
(4)  002014  000170                             .WORD   120.
(5)  002016                             L$HPCP::                ;POINTER TO H.W. QUES.
(4)  002016  034640                             .WORD   L$HARD
(5)  002020                             L$SPCP::                ;POINTER TO S.W. QUES.
(4)  002020  000000                             .WORD   0
(5)  002022                             L$HPTP::                ;PTR. TO DEF. H.W. PTABLE
(4)  002022  002364                             .WORD   L$HW
(5)  002024                             L$SPTP::                ;PTR. TO S.W. PTABLE
(4)  002024  000000                             .WORD   0
(5)  002026                             L$LADP::                ;DIAG. END ADDRESS
(4)  002026  040004                             .WORD   L$LAST
(5)  002030                             L$STA::                 ;RESERVED FOR APT STATS
(4)  002030  000000                             .WORD   0
(5)  002032                             L$CO::
(4)  002032  000000                             .WORD   0
(5)  002034                             L$DTYP::                ;DIAGNOSTIC TYPE
(4)  002034  000000                             .WORD   0
(5)  002036                             L$APT::                 ;APT EXPANSION
(4)  002036  000000                             .WORD   0
(5)  002040                             L$DTP::                 ;PTR. TO DISPATCH TABLE
(4)  002040  002132                             .WORD   L$DISPATCH
(5)  002042                             L$PRIO::                ;DIAGNOSTIC RUN PRIORITY
(4)  002042  000000                             .WORD   0
(5)  002044                             L$ENVI::                ;FLAGS DESCRIBE HOW IT WAS SETUP
(4)  002044  000000                             .WORD   0
(5)  002046                             L$EXP1::                ;EXPANSION WORD
(4)  002046  000000                             .WORD   0
(5)  002050                             L$MREV::                ;SVC REV AND EDIT #
(4)  002050      003                            .BYTE   C$REVISION
(3)  002051      003                            .BYTE   C$EDIT
```

```
 (5)  002052                      L$EF::                          ;DIAG. EVENT FLAGS
 (4)  002052  000000                      .WORD   0
 (5)  002054  000000                      .WORD   0
 (5)  002056                      L$SPC::
 (4)  002056  000000                      .WORD   0
 (5)  002060                      L$DEVP::                        ; POINTER TO DEVICE TYPE LIST
 (4)  002060  003130                      .WORD   L$DVTYP
 (5)  002062                      L$REPP::                        ;PTR. TO REPORT CODE
 (4)  002062  000000                      .WORD   0
 (5)  002064                      L$EXP4::
 (4)  002064  000000                      .WORD   0
 (5)  002066                      L$EXP5::
 (4)  002066  000000                      .WORD   0
 (5)  002070                      L$AUT::                         ;PTR. TO ADD UNIT CODE
 (4)  002070  011364                      .WORD   L$AU
 (5)  002072                      L$DUT::                         ;PTR. TO DROP UNIT CODE
 (4)  002072  011360                      .WORD   L$DU
 (5)  002074                      L$LUN::                         ;LUN FOR EXERCISERS TO FILL
 (4)  002074  000000                      .WORD   0
 (5)  002076                      L$DESP::                        ;POINTER TO DIAG. DESCRIPTION
 (4)  002076  002414                      .WORD   L$DESC
 (5)  002100                      L$LOAD::                        ;GENERATE SPECIAL AUTOLOAD EMT
 (4)  002100  104035                      EMT     E$LOAD
 (5)  002102                      L$ETP::                         ;POINTER TO ERRTBL
 (4)  002102  000000                      .WORD   0
 (5)  002104                      L$ICP::                         ;PTR. TO INIT CODE
 (4)  002104  010570                      .WORD   L$INIT
 (5)  002106                      L$CCP::                         ;PTR. TO CLEAN-UP CODE
 (4)  002106  011354                      .WORD   L$CLEAN
 (5)  002110                      L$ACP::                         ;PTR. TO AUTO CODE
 (4)  002110  011256                      .WORD   L$AUTO
 (5)  002112                      L$PRT::                         ;PTR. TO PROTECT TABLE
 (4)  002112  002122                      .WORD   L$PROT
 (5)  002114                      L$TEST::                        ;TEST NUMBER
 (4)  002114  000000                      .WORD   0
 (5)  002116                      L$DLY::                         ;DELAY COUNT
 (4)  002116  000000                      .WORD   0
 (5)  002120                      L$HIME::                        ;PTR. TO HIGH MEM
 (4)  002120  000000                      .WORD   0
3748
3749
3755
3756  002122                              BGNPROT
 (3)  002122                      L$PROT::
3757  002122  177777                      .WORD   -1
3758  002124  177777                      .WORD   -1
3759  002126  177777                      .WORD   -1
3760  002130                              ENDPROT
3761
```

```
                                .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.
(4) 002130  000114                      .WORD   76
(3) 002132                      L$DISPATCH::
(6) 002132  011366                      .WORD   T1
(6) 002134  011514                      .WORD   T2
(6) 002136  011560                      .WORD   T3
(6) 002140  011746                      .WORD   T4
(6) 002142  012112                      .WORD   T5
(6) 002144  012246                      .WORD   T6
(6) 002146  012376                      .WORD   T7
(6) 002150  012526                      .WORD   T8
(6) 002152  012670                      .WORD   T9
(6) 002154  013054                      .WORD   T10
(6) 002156  013240                      .WORD   T11
(6) 002160  013426                      .WORD   T12
(6) 002162  013576                      .WORD   T13
(6) 002164  013704                      .WORD   T14
(6) 002166  014134                      .WORD   T15
(6) 002170  014364                      .WORD   T16
(6) 002172  014614                      .WORD   T17
(6) 002174  015044                      .WORD   T18
(6) 002176  015340                      .WORD   T19
(6) 002200  015634                      .WORD   T20
(6) 002202  016064                      .WORD   T21
(6) 002204  016314                      .WORD   T22
(6) 002206  016544                      .WORD   T23
(6) 002210  016774                      .WORD   T24
(6) 002212  017224                      .WORD   T25
(6) 002214  017454                      .WORD   T26
(6) 002216  017704                      .WORD   T27
(6) 002220  020134                      .WORD   T28
(6) 002222  020432                      .WORD   T29
(6) 002224  020662                      .WORD   T30
(6) 002226  021226                      .WORD   T31
(6) 002230  021540                      .WORD   T32
(6) 002232  021702                      .WORD   T33
(6) 002234  022044                      .WORD   T34
(6) 002236  022220                      .WORD   T35
(6) 002240  022424                      .WORD   T36
(6) 002242  022570                      .WORD   T37
(6) 002244  022740                      .WORD   T38
(6) 002246  023104                      .WORD   T39
(6) 002250  023272                      .WORD   T40
(6) 002252  023506                      .WORD   T41
(6) 002254  023724                      .WORD   T42
(6) 002256  024056                      .WORD   T43
(6) 002260  024274                      .WORD   T44
(6) 002262  024434                      .WORD   T45
(6) 002264  024640                      .WORD   T46
```

```
(6)  002266  025044                    .WORD   T47
(6)  002270  025250                    .WORD   T48
(6)  002272  025454                    .WORD   T49
(6)  002274  025660                    .WORD   T50
(6)  002276  026064                    .WORD   T51
(6)  002300  026270                    .WORD   T52
(6)  002302  026474                    .WORD   T53
(6)  002304  026702                    .WORD   T54
(6)  002306  027106                    .WORD   T55
(6)  002310  027312                    .WORD   T56
(6)  002312  027516                    .WORD   T57
(6)  002314  027722                    .WORD   T58
(6)  002316  030126                    .WORD   T59
(6)  002320  030332                    .WORD   T60
(6)  002322  030536                    .WORD   T61
(6)  002324  030742                    .WORD   T62
(6)  002326  031146                    .WORD   T63
(6)  002330  031352                    .WORD   T64
(6)  002332  031556                    .WORD   T65
(6)  002334  031762                    .WORD   T66
(6)  002336  032166                    .WORD   T67
(6)  002340  032372                    .WORD   T68
(6)  002342  032576                    .WORD   T69
(6)  002344  033002                    .WORD   T70
(6)  002346  033206                    .WORD   T71
(6)  002350  033412                    .WORD   T72
(6)  002352  033616                    .WORD   T73
(6)  002354  034022                    .WORD   T74
(6)  002356  034226                    .WORD   T75
(6)  002360  034432                    .WORD   T76
3771                                   ;LNT.ED DIFINED AT END OF PROGRAM TO BE LAST TEST NUMBER.
3772
3779
3780
3781
3782
3783
```

```
                                .SBTTL  DEFAULT HARDWARE P-TABLE

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

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

002364  000007                          .WORD   7               ;MICRO CPU TYPE
002366  160170                          .WORD   160170          ;M8200,4,7 CSR ADDRESS
002370  0003C0                          .WORD   300             ;M8200,4,7 VECTOR ADDRESS
002372  005000                          .WORD   5000            ;INTERRUPT PRIORITY LEVEL
002374  000003                          .WORD   3               ;LINE UNIT TYPE
002376  000056                          .WORD   56              ;SWITCH PACK #1 (DDCMP LINE #)
002400  000000                          .WORD   0               ;SWITCH PACK #2 (BM873 BOOT ADDRESS)
002402  000000                          .WORD   0               ;SWITCH PACK #3
002404  000000                          .WORD   0               ;TEST CONNECTOR INSTALLED FLAG
002406  000004                          .WORD   4               ;CONTAINS BAUD RATE 4=56K BAUD DEFAULT
                                                                ;0=2.4K , 1-4.8K , 2-9.6K , 3-19.2K , 4-56K
                                                                ;5-250K , 6=500K , 7-1 MEG BAUD
002410  000000                          .WORD   0               ;0=RUN SW OFF, 1-SW ON
002412                                  ENDHW
(3) 002412                      L10001:
```

```
                                .SBTTL  SOFTWARE P-TABLE

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

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

002414                                  ENDSW
(3)  002414                     L10002:
```

```
                                .SBTTL  GLOBAL EQUATES SECTION

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

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

```
(1)  000036  EF.CONTINUE==  30.   ; CONTINUE COMMAND WAS ISSUED
(1)  000035  EF.NEW==       29.   ; A NEW PASS HAS BEEN STARTED
(1)  000034  EF.PWR==       28.   ; A POWER-FAIL/POWER-UP OCCURRED
(1)                               ;
(1)                               ;
(1)                               ; PRIORITY LEVEL DEFINITIONS
(1)                               ;
(1)  000340                       PRI07== 340
(1)  000300                       PRI06== 300
(1)  000240                       PRI05== 240
(1)  000200                       PRI04== 200
(1)  000140                       PRI03== 140
(1)  000100                       PRI02== 100
(1)  000040                       PRI01== 40
(1)  000000                       PRI00== 0
(1)                               ;
(1)                               ;OPERATOR FLAG BITS
(1)                               ;
(1)  000004                       EVL==     4
(1)  000010                       LOT==    10
(1)  000020                       ADR==    20
(1)  000040                       IDU==    40
(1)  000100                       ISR==   100
(1)  000200                       UAM==   200
(1)  000400                       BOE==   400
(1)  001000                       PNT==  1000
(1)  002000                       PRI==  2000
(1)  004000                       IXE==  4000
(1)  010000                       IBE== 10000
(1)  020000                       IER== 20000
(1)  040000                       LOE== 40000
(1)  100000                       HOE== 100000
3853
3854
3855
3856
3857
3858                              ;*****************************************************************
3859                              ;* INSTRUCTION DEFINITIONS
3860                              ;*****************************************************************
3861  022626                      POP2SP=22626          ;INCREMENT STACK TWICE
3862
3863
3864
3865                              ;;****************************************************************
3866                              ;* PROGRAM EVENT FLAG DEFINITIONS
3867                              ;;****************************************************************
3868
3869
3870
```

```
                                .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  0000E2
                                        .EVEN

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

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

                                ;**********************************************************************
                                ;* MISCELLANEOUS STORAGE
                                ;**********************************************************************
002550  000000                  $TMPO:  .WORD   0               ;SCRATCH STORAGE
002552  000000                  LOGDEV: .WORD   0               ;LOGICAL DEVICE NUMBER
002554  000000                  PSTACK: .WORD   0               ;BASE LEVEL PROGRAM STACK POINTER
002556  000000                  SUBRPC: .WORD   0               ;PC OF SUBR CALL FOR ERROR REPORTS
002560  000000                  ERRFLG: .WORD   0               ;SUBROUTINE ERROR FLAG
002562  000000                  RETADR: .WORD   0               ;SUBR ERROR RETURN ADDRESS
002564  000000                  STRTSW: .WORD   0               ;SWITCHES AT START OF PROGRAM
002566  000000                  STAT:   .WORD   0               ;M8200,4,7 STATUS WORD STORAGE
002570  000000                  CLKX:   .WORD   0               ;
002572  000000                  MASKX:  .WORD   0               ;
002574  000000                  SAVSP:  .WORD   0               ;STACK POINTER STORAGE
002576  000000                  SAVPC:  .WORD   0               ;PROGRAM COUNTER STORAGE
002600  000000                  ZERO:   .WORD   0               ;
002602  000001                  ONE:    .WORD   1               ;
002604  037776                  MEMLIM: .WORD   MEMEND          ;HIGHEST LOCATION FOR 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
```

```
3922  002620  000000                  FLAG:   .WORD   0               ;SCRATCH STORAGE
3923  002622  000000                  RUN:    .WORD   0               ;POINTER TO RUNNING DEVICES
3924  002624  000000                  MRO:    .WORD   0
3925  002626  000000                  WTYPE:  .WORD   0
3926  002630  000000                  TYPE:   .WORD   0
3927  002632  000000                  $GDADR: .WORD   0               ;CONTAINS ADDRESS OF 'GOOD' DATA
3928  002634  000000                  $BDADR: .WORD   0               ;CONTAINS ADDRESS OF 'BAD' DATA
3929  002636  000000                  $GDDAT: .WORD   0               ;CONTAINS 'GOOD' DATA
3930  002640  000000                  $BDDAT: .WORD   0               ;CONTAINS 'BAD' DATA
3931  002642  000000                          .WORD   0               ;RESERVED--NOT TO BE USED
3932  002644  000000                          .WORD   0
3933  002646  000000                  FTIME:  .WORD   0
3934  002650  000000                  SAVE4:  .WORD   0
3935  002652  000000                  SAVE6:  .WORD   0
3936
3937                                  ;*****************************************************************************
3938                                  ;* DATA PATTERNS
3939                                  ;*****************************************************************************
3940  002654    000   377   000       MEMDAT: .BYTE   0,-1,0,-1,125,252,125,252
      002657    377   125   252
      002662    125   252
3941  002664    000   000   377       SPDAT:  .BYTE   0,0,-1,-1,125,125,252,252
      002667    377   125   125
      002672    252   252
3942                                          .EVEN
3943
3944                                  ;*****************************************************************************
3945                                  ;* PROGRAM CONTROL FLAGS
3946                                  ;*****************************************************************************
3947  002674      000                 INIFLG: .BYTE   0               ;PROGRAM INITIALIZING FLAG
3948        002676                            .EVEN
3949  002676      000                 LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
3950  002677      000                 QV.FLG: .BYTE   0               ;QUICK VERIFY FLAG
3951                                          .EVEN
3952
3953                                  ;*****************************************************************************
3954                                  ;* DEFINITION OF M8200,4,7 STATUS WORDS - STAT1,STAT2,STAT3
3955                                  ;*
3956                                  ;*
3957                                  ;*      STAT1 - BITS 00-08 IS M8200,4,7 VECTOR ADDRESS
3958                                  ;*              BIT15-1 LINE UNIT IS AN M8203
3959                                  ;*              BIT14=0 NO TEST CONNECTOR(S) USED
3960                                  ;*              BIT14=1 H-XXX TEST CONNECTOR WILL BE USED
3961                                  ;*              BIT13=0 LINE UNIT IS AN M8201
3962                                  ;*              BIT13=1 LINE UNIT IS AN M8202
3963                                  ;*              BIT12=1 NO LINE UNIT
3964                                  ;*              BITS 09-11 IS M8200,4,7 PRIORITY LEVEL
3965                                  ;*
3966                                  ;*      STAT2 - LOW BYTE IS SWITCH PACK #1 (DDCMP LINE NUMBER)
3967                                  ;*              HIGH BYTE IS SWITCH PACK #2 (BM873 BOOT ADDRESS)
3968                                  ;*
3969                                  ;*      STAT3 - BIT0 1 DO FREE RUNNING TESTS ON M8200,4,7
3970                                  ;*
3971                                  ;*****************************************************************************
3972  002700  000000                  STAT1:  .WORD   0
3973  002702  000000                  STAT2:  .WORD   0
```

```
3974  002704  000000                  STAT3:  .WORD   0
3975
3976                                  ;*********************************************************************
3977                                  ;* POINTERS TO M8200,4,7 VECTORS AND REGISTERS
3978                                  ;*********************************************************************
3979  002706  000000                  KMRVEC: 0       ;POINTER TO M8200,4,7 RCV INTRPT VECTOR
3980  002710  000000                  KMRLVL: 0       ;POINTER TO M8200,4,7 RCV INTRPT SERVICE PS
3981  002712  000000                  KMTVEC: 0       ;POINTER TO M8200,4,7 TX INTRPT VECTOR
3982  002714  000000                  KMTLVL: 0       ;POINTER TO M8200,4,7 TX INTRPT SERVICE PS
3983  002716  000000                  KMCSR:  0       ;POINTER TO M8200,4,7 CONTROL STATUS REGISTER
3984  002720  000000                  KMCSRH: 0       ;POINTER TO M8200,4,7 CONTROL STATUS REGISTER HIGH BYTE
3985  002722  000000                  KMCTL:  0       ;POINTER TO M8200,4,7 CONTROL OUT REGISTER
3986  002724  000000                  KMP04:  0
3987  002726  000000                  KMP06:  0       ;POINTER TO M8200,4,7 PORT REGISTER - SEL6
3988
3989                                  ;;****  PRIMARY REG ADRS STORAGE FOR THIS UNIT  ******
3990                                  ;THESE LOCATIONS WILL BE LOADED FOR THE CURRENT UNIT, IN INIT CODE
3991  002730                          REGADR:
3992
3993                                  ;;****  STACK USED FOR SUBROUTINE LINKAGE  *****
3994  002730  000100                          .BLKW   100
3995  003130                          SSTACK:
3996
3997
3998
3999
4000
4001
4002
```

```
 4004                                          .SBTTL  GLOBAL TEXT SECTION
 4005
 4006                                          ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 4007                                          ;%       THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
 4008                                          ;%       MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
 4009                                          ;%       MORE THAN ONF TEST.
 4010                                          ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 4011
 4012                                          ;;***************************************************************************
 4013                                          ;* NAMES OF DEVICES SUPPORTED BY PROGRAM
 4014                                          ;;***************************************************************************
 4015  003130                                          DEVTYP  <M8200,M8204,OR M8207>
  (4)  003130                                  L$DVTYP::
  (3)  003130  034115  030062  026060                  .ASCIZ  /M8200,M8204,OR M8207/
  (3)  003136  034115  030062  026064
  (3)  003144  051117  046440  031070
  (3)  003152  033460     000
  (2)          003156                                  .EVEN
 4016
 4017
 4018                                          ;
 4019                                          ; FORMAT STATEMENTS USED IN PRINT CALLS
 4020                                          ;
 4021
 4028
 4029
 4030
 4031
 4032
```

```
4034                            .SBTTL  GLOBAL SUBROUTINES
4035
4036                            ;////////////////////////////////////////////////////////////////////////////
4037                            ;/      THE GLOBAL SUBROUTINES ARE CALLED BY MORE THAN ONE TEST
4038                            ;////////////////////////////////////////////////////////////////////////////
4039
4040
4041
4042                            ;-------------------------------------------------------------------
4043                            ; MACRO'S NEEDED TO CALL SUBROUTINES
4044                            ;-------------------------------------------------------------------
4045
4046                            .MACRO  ERROR,XYX,ZZ
4047                                    MOV     R4,$BDDAT
4048                                    .IF     B ZZ
4049                                    MOV     R2,$GDDAT
4050                                    .ENDC
4051                                    MOV     MR0,$BDADR
4052                                    ERRDF   XYX',EM'XYX',ERR'XYX'
4053                            .ENDM
4054                            .MACRO  RERROR  XXX
4055                                    MOV     R4,$BDDAT
4056                                    CLRB    $BDDAT+1
4057                                    CLRB    $GDDAT+1
4058                                    MOV     R2,$GDADR
4059                                    ERRDF   XXX',EM'XXX',ERR'XXX'
4060                            .ENDM
4061                            .MACRO  BERROR  XXX
4062                                    MOV     R4,$BDDAT
4063                                    MOV     R5,$GDDAT
4064                                    CLRB    $BDDAT+1
4065                                    CLRB    $GDDAT+1
4066                                    ERRDF   XXX',EM'XXX',ERR'XXX'
4067                            .ENDM
4068                            .MACRO  ED$CALL XY
4069                                    .LIST
4070                                    ;****************** TEST 'XY' ******************
4071                                    .NLIST
4072                                    .ENDM
4073                                    .MACRO  BADHEAD
4074                                    .RADIX  10
4075                                    ED$CALL \T$TESTNUM+1
4076                                    .RADIX  8
4077                                    .ENDM
4078                                    .MACRO  K40NLY  ?N2
4079                                    .LIST
4080                                    ;DON'T DO TEST IF M8200 OR M8204
4081                                    .NLIST
4082                                    CMP     MEMSZ,#2000
4083                                    BNE     N2
4084                                    EXIT    TST
4085                            N2:
4086                                    .ENDM
4087                            .MACRO  MYINT
4088                                    .LIST
4089                                    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,@KMCSRH   ;SET INST.
003164  142777  000300  177526          BICB    #BIT6!BIT7,@KMCSRH
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,@KMCSRH
003252  012577  177450                  MOV     (R5)+,@KMP06
003256  152777  000003  177434          BISB    #BIT1!BIT0,@KMCSRH
003264  142777  000007  177426          BICB    #BIT2!BIT1.BIT0,@KMCSRH
003272  000205                          RTS     R5

003274                          CLRALL:
                                        ;CLEARS C & Z BITS AND BR
003274                                  ROMCLK
 (1)    003274  004537  003244          JSR     R5,..ROMCLK                     ;CLOCK INSTRUCTION
003300  000400                          400                             ;0 TO BR
003302                                  ROMCLK
 (1)    003302  004537  003244          JSR     R5,..ROMCLK                     ;CLOCK INSTRUCTION
003306  063220                          63220                           ;SP(0) TO BR
003310                                  ROMCLK
 (1)    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
 (1)    003320  004537  003244          JSR     R5,..ROMCLK                     ;CLOCK INSTRUCTION
003324  000401                          401                             ;1 TO BR
003326  000207                          RTS     PC
```

```
4142
4143  003330                          SETBR1:
4144                                          ;THIS SUBROUTINE SETS BR1 BIT
4145
4146  003330                          ROMCLK                  ;NEXT WORD IS INSTRUCTION
 (1)  003330  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4147  003334  000402                          000402                  ;BR_002
4148  003336  000207                          RTS     PC
4149
4150  003340                          SETBR4:
4151                                          ;THIS SUBROUTINE SETS BR4 BIT
4152
4153  003340                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION
 (1)  003340  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4154  003344  000402                          402
4155  003346  000207                          RTS     PC
4156
4157  003350                          SETBR7:
4158                                          ;THIS SUBROUTINE SETS BR7 BIT
4159
4160  003350                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION
 (1)  003350  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4161  003354  000600                          600
4162  003356  000207                          RTS     PC
4163
4164
4165  003360                          SETZ:
4166                                          ;THIS SUBROUTINE SETS THE Z BIT
4167
4168  003360                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION
 (1)  003360  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4169  003364  000777                          000777                  ;BR_377
4170  003366  000207                          RTS     PC
4171
4172  003370                          RAMDAT:
4173                                          ;THIS SUBROUTINE LOADS R4 WITH THE LOWEST
4174                                          ;8 BITS OF THE CRAM PC.
4175
4176  003370  017605  000000                  MOV     @(SP),R5        ;GOOD DATA
4177  003374  062716  000002                  ADD     #2,(SP)         ;ADJUST STACK
4178  003400  005011                          CLR     (R1)            ;CLEAR BIT10
4179  003402  052711  000400                  BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION IN CRAM THAT
4180                                                                  ;JUMPED TO, IT LOADS BR WITH IT
4181  003406  005011                          CLR     (R1)            ;CLR BIT8
4182  003410                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION
 (1)  003410  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
4183  003414  061225                          061225                  ;MOV BR TO PORT 5
4184  003416  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND' IN R4
4185  003422  000207                          RTS     PC              ;RETURN
4186
4187
4188  003424                          MEMSET:
4189                                          ;THIS SUBROUTINE LOADS CRAM WITH SPECIAL INSTRUCTIONS
4190                                          ;FOR THE CRAM JUMP TEST. ALL CRAM LOCATIONS ARE LOADED
4191                                          ;WITH INSTRUCTIONS THAT MOVE A 37 TO THE BR, EXCEPT THE
4192                                          ;FOLLOWING CRAM ADDRESSES: 0,1,4,7,525,1777. THESE LOCATIONS
```

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

003424  005000                  CLR     R0              ;R0 = CRAM ADDRESS
003426  012711  002000          1$:     MOV     #BIT10,(R1)     ;SET ROMO
003432  010061  000004                  MOV     R0,4(R1)        ;LOAD CRAM ADDRESS
003436  012761  000437  000006          MOV     #437,6(R1)      ;LOAD INSTRUCTION
003444  052711  020000                  BIS     #BIT13,(R1)     ;WRITE INSTRUCTION IN CRAM
003450  005200                  INC     R0              ;NEXT ADDRESS
003452  022700  002000          CMP     #2000,R0        ;DONE YET?
003456  001363                  BNE     1$              ;BR IF NO
003460  005000                  CLR     R0              ;INDEX REGISTER
003462  012711  002000          2$:     MOV     #BIT10,(R1)     ;SET ROMO
003466  016061  003522  000004          MOV     CRAMA(R0),4(R1) ;LOAD CRAM ADDRESS IN SEL4
003474  016061  003536  000006          MOV     INSTU(R0),6(R1) ;LOAD INSTRUCTION TO BE WRITTEN
003502  052711  020000                  BIS     #BIT13,(R1)     ;WRITE CRAM!
003506  005720                  TST     (R0)+           ;NEXT
003510  022700  000014          CMP     #14,R0          ;DONE YET?
003514  001362                  BNE     2$              ;BR IF NO
003516  005011                  CLR     (R1)            ;CLEAR ALL BITS
003520  000207                  RTS     PC              ;RETURN

003522  000000  000001  000004  CRAMA:  .WORD   0,1,4,7,1777,525
003530  000007  001777  000525

003536  000400                  INSTU:  000400          ;BR_0
003540  000401                          000401          ;BR_1
003542  000404                          000404          ;BR_4
003544  000407                          000407          ;BR_7
003546  000777                          000777          ;BR_377
003550  000525                          000525          ;BR_125

003552                          SETVEC:
                                        ;THIS SUBROUTINE LOADS THE VECTORS AND VECTOR LEVELS

003552  012577  177130                  MOV     (R5)+,@KMRVEC   ;LOAD BASE VECTOR
003556  012577  177130                  MOV     (R5)+,@KMTVEC   ;LOAD VECTOR + 2
003562  012577  177122                  MOV     (R5)+,@KMRLVL   ;LOAD VECTOR + 4
003566  012577  177122                  MOV     (R5)+,@KMTLVL   ;LOAD VECTOR + 6
003572  000205                  RTS     R5              ;RETURN


003574                          NPRSET:
                                        ;THIS SUBROUTINE LOADS IBUS REGISTERS 0-7
                                        ;WITH NPR INFORMATION (INBA, OUTBA, OUT DATA)

003574  010246                  MOV     R2,-(SP)        ;SAVE R2
003576  005002                  CLR     R2              ;START AT IBUS REG 0
003600  112561  000004          1$:     MOVB    (R5)+,4(R1)     ;LOAD PORT4
003604  042737  000017  003622          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
003612  050237  003622                  BIS     R2,2$           ;ADD ADDRESS TO INSTRUCTION
003616                          ROMCLK
 (1)    003616  004537  003244          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
003622  122100                  2$:     122100                  ;MOVE PORT4 TO IBUS REG
003624  005202                  INC     R2              ;NEXT ADDRESS
003626  022702  000010          CMP     #10,R2          ;ALL DONE?
```

```
4247  003632  001362                    BNE     1$              ;BR IF NO
4248  003634  012602                    MOV     (SP)+,R2        ;RESTORE R2
4249  003636  000205                    RTS     R5              ;RETURN
4250
4251
4252  003640                    MEMLD:
4253                            ;THIS SUBROUTINE LOADS THE FIRST 8 LOCATIONS OF MAIN
4254                            ;MEMORY WITH THIS DATA: 0,-1,,0,-1,125,252,125,252
4255
4256  003640  013637  002550            MOV     @(SP)+,$TMP0    ;PUT POINTER TO DATA IN R0
4257  003644  062746  000002            ADD     #2,-(SP)        ;ADJUST STACK
4258
4259  003650  013700  002550    MEMLD2: MOV     $TMP0,R0        ;GET ADDR.
4260  003654  012704  000010            MOV     #10,R4          ;DO 8 LOADS
4261  003660                            ROMCLK
 (1)  003660  004537  003244            JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
4262  003664  010000                    010000                  ;MAR < 0
4263  003666                            ROMCLK                  ;CLR    MAR HI
 (1)  003666  004537  003244            JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
4264  003672  004000                    004000
4265  003674  112077  177024    1$:     MOVB    (R0)+,@#KMP04   ;LOAD PORT4
4266  003700                            ROMCLK
 (1)  003700  004537  003244            JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
4267  003704  136500                    136500                  ;MOV DATA TO MEM, AUTO INC MAR
4268  003706  005304                    DEC     R4              ;DECREMENT COUNT
4269  003710  001371                    BNE     1$              ;BR IF NOT DONE
4270
4271  003712                            ROMCLK                  ;LOAD MEM ADDR. 0
 (1)  003712  004537  003244            JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
4272  003716  010000                    10000
4273  003720  012703  000010            MOV     #10,R3          ;CHECK 8. MEM LOCS.
4274  003724  013700  002550            MOV     $TMP0,R0
4275  003730                    2$:     ROMCLK                  ;READ FROM MEM,PUT INTO PORT 4
 (1)  003730  004537  003244            JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
4276  003734  055224                    55224
4277
4278  003736  112037  002636            MOVB    (R0)+,$GDDAT    ;EXPECTED.
4279  003742  117704  176756            MOVB    @#KMP04,R4      ;RECIEVED.
4280  003746  123704  002636            CMPB    $GDDAT,R4       ;OK?
4281  003752  001414                    BEQ     3$
4282  003754                            ERROR   36
 (5)  003772  104455                    TRAP    C$ERDF
 (6)  003774  000044                    .WORD   36
 (6)  003776  005640                    .WORD   EM36
 (6)  004000  010432                    .WORD   ERR36
4283  004002  000402                    BR      4$
4284  004004  005303            3$:     DEC     R3              ;CHECKED ALL?
4285  004006  001350                    BNE     2$              ;NO-DO NEXT ONE.
4286  004010                    4$:
4287  004010  000207                    RTS     PC              ;RETURN
4288
4289
4292  004012                    SPLD:
4293                            ;THIS SUBROUTINE LOADS THE FIRST 8 SCRATCH PAD
4294                            ;LOCATIONS WITH: 0,0,-1,-1,125,125,252,252
4295
```

```
4296  004012  013600                        MOV     @(SP)+,R0       ;PUT POINTER TO DATA IN R5
4297  004014  062746  000002                ADD     #2,-(SP)        ;ADJUST STACK
4298  004020  005004                        CLR     R4              ;START AT SP ADDRESS 0
4299  004022  112077  176676        1$:     MOVB    (R0)+,@KMP04    ;LOAD PORT4 WITH DATA
4300  004026  042737  000017  004044        BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
4301  004034  050437  004044                BIS     R4,2$           ;ADD ADDRESS TO INSTRUCTION
4302  004040                                ROMCLK
 (1)  004040  004537  003244                JSR     R5,,ROMCLK              ;CLOCK INSTRUCTION
4303  004044  123100                2$:     123100                  ;MOVE DATA TO SP
4304  004046  005204                        INC     R4              ;INCREMENT COUNT
4305  004050  022704  000010                CMP     #10,R4          ;DONE YET?
4306  004054  001362                        BNE     1$              ;BR IF NO
4307  004056  000207                        RTS     PC              ;RETURN
4308
4309
4310  004060                        CLRC:
4311                                        ;THIS SUBROUTINE CLEARS THE MICRO PROCESSOR C BIT
4312
4313  004060                                ROMCLK
 (1)  004060  004537  003244                JSR     R5,,ROMCLK              ;CLOCK INSTRUCTION
4314  004064  010000                        010000                  ;MAR_0
4315  004066                                ROMCLK
 (1)  004066  004537  003244                JSR     R5,,ROMCLK              ;CLOCK INSTRUCTION
4316  004072  040400                        040400.<0*20>           ;CLEAR C BIT
4317  004074  000207                        RTS     PC              ;RETURN
4318
4319
4320  004076                        SETC:
4321                                        ;THIS SUBROUTINE SETS THE MICRO PROCESSOR C BIT
4322
4323  004076                                ROMCLK
 (1)  004076  004537  003244                JSR     R5,,ROMCLK              ;CLOCK INSTRUCTION
4324  004102  010003                        010003                  ;MAR_3
4325  004104                                ROMCLK
 (1)  004104  004537  003244                JSR     R5,,ROMCLK              ;CLOCK INSTRUCTION
4326  004110  040403                        040403!<0*20>           ;SET C BIT
4327  004112  000207                        RTS     PC              ;RETURN
4328
4329
4330
4331
```

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

```
4340
4341
4342
4343  004114  047045  052045  047045  FM1:    .ASCIZ  /%N%T%N/
      004122  000
4344  004123  045     031517  051445  TFM1:   .ASCIZ  /%03%S5%03%S5%03%N2/
      004130  022465  031517  051445
      004136  022465  031517  047045
      004144  000062
4345  004146  047445  022466  031123  TFM2:   .ASCIZ  /%06%S2%06%N2/
      004154  047445  022466  031116
      004162  000
4346  004163  045     031517  051445  TFM5:   .ASCIZ  /%03%S5%03%N2/
      004170  022465  031517  047045
      004176  000062
4347  004200  047045  047445  022463  TFM27:  .ASCIZ  /%N%03%S5%06%S7%06%N2/
      004206  032523  047445  022466
      004214  033523  047445  022466
      004222  031116  000
4348  004225  045     022516  043101  TFM37:  .ASCIZ  /%N%AFAILING ADDRESS IS: %06/
      004232  044501  044514  043516
      004240  040440  042104  042522
      004246  051523  044440  035123
      004254  022440  033117  000
4349
4350
```

```
4352
4353
4354  004261    122  043505  051511  EM1:    .ASCIZ  &REGISTER ADDRESS TEST&
      004266 042524  020122  042101
      004274 051104  051505  020123
      004302 042524  052123     000
4355  004307    111  052502  025123  EM2:    .ASCIZ  &!BUS* REGISTER DUAL ADDRESSING TEST&
      004314 051040  043505  051511
      004322 042524  020122  052504
      004330 046101  040440  042104
      004336 042522  051523  047111
      004344 020107  042524  052123
      004352    000
4356  004353    111  052502  020123  EM30:   .ASCIZ  ''IBUS REGISTER DUAL ADDRESSING TEST''
      004360 042522  044507  052123
      004366 051105  042040  040525
      004374 020114  042101  051104
      004402 051505  044523  043516
      004410 052040  051505  000124
4357  004416 051102  051040  043505  EM3:    .ASCIZ  /BR REGISTER DATA TEST/
      004424 051511  042524  020122
      004432 040504  040524  052040
      004440 051505  000124
```

```
 4359  004444  041523  040522  041524  EM4:    .ASCIZ  /SCRATCH PAD DATA TEST/
       004452  020120  040520  020104
       004460  040504  040524  052040
       004466  051505  000124
```

```
 4361  004472  041523  040522  041524  EMS:    .ASCIZ  /SCRATCH PAD DUAL ADDRESSING TEST/
       004500  020110  040520  020104
       004506  052504  046101  040440
       004514  042104  042522  051523
       004522  047111  020107  042524
       004530  052123     000
```

```
4363  004533     115  044501  020116  EM6:    .ASCIZ  /MAIN MEMORY DATA TEST/
      004540  042515  047515  054522
      004546  042040  052101  020101
      004554  042524  052123     000
4364  004560     115  044501  020116  EM7:    .ASCIZ  /MAIN MEMORY DUAL ADDRESSING TEST/
      004566  042515  047515  054522
      004574  042040  040525  020114
      004602  042101  051104  051505
      004610  044523  043516  032040
      004616  051505  000124
```

```
 4366  004622  052501  047524  046440  EM10:   .ASCIZ  /AUTO MARINC FUNCTION TEST/
       004630  051101  047111  020103
       004636  052506  041516  044524
       004644  047117  052040  051505
       004652  000124
 4367  004654  050116  020122  042524  EM11:   .ASCIZ  /NPR TEST/
       004662  052123     000
 4368  004665     115  046125  044524  EM12:   .ASCIZ  /MULTIPLE NPR TEST/
       004672  046120  020105  050116
       004700  020122  042524  052123
       004706     000
 4369  004707     116  047117  042440  EM13:   .ASCIZ  /NON EX MEM FAILED/
       004714  020130  042515  020115
       004722  040506  046111  042105
       004730     000
 4370  004731     120  047522  051107  EM14:   .ASCIZ  /PROGRAM CLOCK TEST/
       004736  046501  041440  047514
       004744  045503  052040  051505
       004752  000124
 4371  004754  046101  020125  052506  EM15:   .ASCIZ  /ALU FUNCTION WITH C BIT CLEAR TEST/
       004762  041516  044524  047117
       004770  053440  052111  020110
       004776  020103  044502  020124
       005004  046103  040505  020122
       005012  042524  052123     000
```

```
 4373  005017     120  053517  051105  EM16:   .ASCIZ  /POWER FAIL: BUS INIT WAS NOT BLOCKED/
       005024  043040  044501  035114
       005032  041040  051525  044440
       005040  044516  020124  040527
       005046  020123  047516  020124
       005054  046102  041517  042513
       005062  000104
 4374  005064                          EM35:
 4375  005064  047506  041522  020105  EM17:   .ASCIZ  /FORCE POWER FAIL ERROR/
       005072  047520  042527  020122
       005100  040506  046111  042440
       005106  051122  051117     000
 4376  005113     116  044517  042523  EM20:   .ASCIZ  /NOISE TEST ON IBUS*,IBUS,SPAD,MEMORY/
       005120  052040  051505  020124
       005126  047117  044440  052502
       005134  025123  044454  052502
       005142  026123  050123  042101
       005150  046454  046505  051117
       005156  000131
 4377  005160  046101  020125  020103  EM21:   .ASCIZ  /ALU C BIT TEST FAILURE/
       005166  044502  020124  042524
       005174  052123  043040  044501
       005202  052514  042522     000
 4378  005207     124  046511  020105  EM22:   .ASCIZ  /TIME OUT ERROR/
       005214  052517  020124  051105
       005222  047522  000122
 4379  005226  046101  020125  052506  EM23:   .ASCIZ  /ALU FUNCTION TEST WITH C BIT SET/
       005234  041516  044524  047117
       005242  052040  051505  020124
       005250  044527  044124  041440
       005256  041040  052111  051440
       005264  052105     000
 4380  005267     125  041520  051440  EM24:   .ASCIZ  /UPC SEQUENCE ERROR/
       005274  050505  042525  041516
       005302  020105  051105  047522
       005310  000122
 4381  005312  050125  043040  044501  EM31:   .ASCIZ  'UP FAILED TO INTERRUPT'
       005320  042514  020104  047524
       005326  044440  052116  051105
       005334  052522  052120     000
 4382  005341     125  020120  047111  EM32:   .ASCIZ  'UP INTERRUPTED TO WRONG VECTOR'
       005346  042524  051122  050125
       005354  042524  020104  047524
       005362  053440  047522  043516
       005370  053040  041505  047524
       005376  000122
 4383  005400  047125  054105  042520  EM33:   .ASCIZ  'UNEXPECTED INTERRUPT FROM UP'
       005406  052103  042105  044440
       005414  052116  051105  052522
       005422  052120  043040  047522
       005430  020115  050125     000
 4384  005435     101  052514  043040  EM34:   .ASCIZ  'ALU FLAG TEST'
       005442  040514  020107  042524
       005450  052123     000
 4385  005453     110  046105  020114  EM25:   .ASCIZ  /HELL RAISER TEST/
       005460  040522  051511  051105
```

```
                005466  052040  051505  000124
4386            005474  040515  047111  040524  EM26:   .ASCIZ  /MAINTANCE REGISTER ERROR/
                005502  041516  020105  042522
                005510  044507  052123  051105
                005516  042440  051122  051117
                005524     000
4387            005525     111  052502  025123  EM27:   .ASCIZ  '!BUS* WRITE/READ ERROR'
                005532  053440  044522  042524
                005540  051057  040505  020104
                005546  051105  047522  000122
4388            005554  047111  052123  052522  EM28:   .ASCIZ  /INSTRUCTION TES' FAILURE/
                005562  052103  047511  020116
                005570  042524  052123  043040
                005576  044501  052514  042522
                005604     000
4389            005605     111  052502  027523  EM29:   .ASCIZ  'IBUS/OBUS WRITE/READ ERROR'
                005612  041117  051525  053440
                005620  044522  042524  051057
                005626  040505  020104  051105
                005634  047522  000122
4390
4391            005640  047511  020120  040515  EM36:   .ASCIZ  'IOP MAIN MEM. LOAD ERROR-RUN MCPU MEM. DIAG.'
                005646  047111  046440  046505
                005654  020056  047514  042101
                005662  042440  051122  051117
                005670  051055  047125  046440
                005676  050103  020125  042515
                005704  027115  042040  040511
                005712  027107     000
4392            005715     000          EM37:   .ASCIZ  //
4393
4394
```

4396

```
 4398  005716      00C                 DH0:    .ASCIZ  //
```

```
4400  005717  000107  047517  020104  DH1:   .ASCIZ  /GOOD    BAD     REGISTER/
      005724  020040  041040  042101
      005732  020040  020040  051040
      005740  043505  051511  042524
      005746  000122
4401  005750  047507  042117  020040  DH2:   .ASCIZ  /GOOD    BAD/
      005756  020040  040502  000104
4402  005764  047507  042117  020040  DH3:   .ASCIZ  /GOOD    BAD     ADDRESS/
      005772  020040  040502  020104
      006000  020040  020040  042101
      006006  051104  051505  000123
4403  006014  047507  042117  020040  DH4:   .ASCIZ  /GOOD    BAD/
      006022  020040  040502  000104
4404  006030  042522  027107  042440  DH27:  .ASCIZ  /REG. EXPECTED FOUND/
      006036  050130  041505  042524
      006044  020104  047506  047125
      006052  000104
4405
4406
4407
4408
4409                          .EVEN
```

4411
4412
4413

```
4415                    ;-------------------------------------------------------------------------------
4416                    ; MACRO'S NEEDED TO REPORT ERRORS
```

```
4418                            ;-----------------------------------------------------------------------
4419                            .MACRO  MDT0
4420                            .ENDM
4421
4422                            .MACRO  MDT1
4423                                    PRINTB #TFM1,$GDDAT,$BDDAT,$GDADR
4424                            .ENDM
4425
4426                            .MACRO MDT2
```

```
4428                                    PRINTB #TFM2,$GDDAT,$BDDAT
4429                            .ENDM
4430
4431                            .MACRO  MDT5
4432                                    PRINTB  #TFM5,$GDDAT,$BDDAT
4433                            .ENDM
```

```
4435                    .MACRO  MDT27
4436                            PRINTB #TFM27,MRO,$GDDAT,$BDDAT
4437                    .ENDM
4438
4439                    .MACRO  $MD,ERNB,ERHM,ERFM
4440                    .NLIST
4441                    ; ERNB = ERROR NUMBER
4442                    ; ERFM = FORMAT NUMBER
4443                    ; ERHM = HEADER NUMBER
4444                    .LIST
4445                            BGNMSG  ERR'ERNB'
4446                            PRINTB  #FM11,#DH'ERHM'
4447                            MDT'ERFM'
4448                            ENDMSG
```

```
4450                                    .ENDM
4451
4452
4453
4454
4455  006054                                    $MD     1,2,2
 (4)  006054                            ERR1::
 (9)  006054  012746  005750                    MOV     #DH2,-(SP)
 (8)  006060  012746  004114                    MOV     #FM1,-(SP)
 (7)  006064  012746  000002                    MOV     #2,-(SP)
 (4)  006070  010600                            MOV     SP,R0
 (5)  006072  104414                            TRAP    C$PNTB
 (5)  006074  062706  000006                    ADD     #6,SP
 (11) 006100  013746  002640                    MOV     $BDDAT,-(SP)
 (10) 006104  013746  002636                    MOV     $GDDAT,-(SP)
 (9)  006110  012746  004146                    MOV     #TFM2,-(SP)
 (8)  006114  012746  000003                    MOV     #3,-(SP)
 (5)  006120  010600                            MOV     SP,R0
 (6)  006122  104414                            TRAP    C$PNTB
 (6)  006124  062706  000010                    ADD     #10,SP
 (4)  006130                            L10003:
 (4)  006130  104423                            TRAP    C$MSG
4456  006132                                    $MD     2,2,2
 (4)  006132                            ERR2::
 (9)  006132  012746  005750                    MOV     #DH2,-(SP)
 (8)  006136  012746  004114                    MOV     #FM1,-(SP)
 (7)  006142  012746  000002                    MOV     #2,-(SP)
 (4)  006146  010600                            MOV     SP,R0
 (5)  006150  104414                            TRAP    C$PNTB
 (5)  006152  062706  000006                    ADD     #6,SP
 (11) 006156  013746  002640                    MOV     $BDDAT,-(SP)
 (10) 006162  013746  002636                    MOV     $GDDAT,-(SP)
 (9)  006166  012746  004146                    MOV     #TFM2,-(SP)
 (8)  006172  012746  000003                    MOV     #3,-(SP)
 (5)  006176  010600                            MOV     SP,R0
 (6)  006200  104414                            TRAP    C$PNTB
 (6)  006202  062706  000010                    ADD     #10,SP
 (4)  006206                            L10004:
 (4)  006206  104423                            TRAP    C$MSG
4457  006210                                    $MD     3,2,2
 (4)  006210                            ERR3::
 (9)  006210  012746  005750                    MOV     #DH2,-(SP)
 (8)  006214  012746  004114                    MOV     #FM1,-(SP)
 (7)  006220  012746  000002                    MOV     #2,-(SP)
 (4)  006224  010600                            MOV     SP,R0
 (5)  006226  104414                            TRAP    C$PNTB
 (5)  006230  062706  000006                    ADD     #6,SP
 (11) 006234  013746  002640                    MOV     $BDDAT,-(SP)
 (10) 006240  013746  002636                    MOV     $GDDAT,-(SP)
 (9)  006244  012746  004146                    MOV     #TFM2,-(SP)
 (8)  006250  012746  000003                    MOV     #3,-(SP)
 (5)  006254  010600                            MOV     SP,R0
 (6)  006256  104414                            TRAP    C$PNTB
 (6)  006260  062706  000010                    ADD     #10,SP
 (4)  006264                            L10005:
 (4)  006264  104423                            TRAP    C$MSG
```

```
4458  006266                                    $MD     4,1,1
 (4)  006266                              ERR4::
 (9)  006266  012746  005717                      MOV     #DH1,-(SP)
 (8)  006272  012746  004114                      MOV     #FM1,-(SP)
 (7)  006276  012746  000002                      MOV     #2,-(SP)
 (4)  006302  010600                              MOV     SP,R0
 (5)  006304  104414                              TRAP    C$PNTB
 (5)  006306  062706  000006                      ADD     #6,SP
(12)  006312  013746  002632                      MOV     $GDADR,-(SP)
(11)  006316  013746  002640                      MOV     $BDDAT,-(SP)
(10)  006322  013746  002636                      MOV     $GDDAT,-(SP)
 (9)  006326  012746  004123                      MOV     #TFM1,-(SP)
 (8)  006332  012746  000004                      MOV     #4,-(SP)
 (5)  006336  010600                              MOV     SP,R0
 (6)  006340  104414                              TRAP    C$PNTB
 (6)  006342  062706  000012                      ADD     #12,SP
 (4)  006346                              L10006:
 (4)  006346  104423                              TRAP    C$MSG
4459  006350                                    $MD     5,1,1
 (4)  006350                              ERR5::
 (9)  006350  012746  005717                      MOV     #DH1,-(SP)
 (8)  006354  012746  004114                      MOV     #FM1,-(SP)
 (7)  006360  012746  000002                      MOV     #2,-(SP)
 (4)  006364  010600                              MOV     SP,R0
 (5)  006366  104414                              TRAP    C$PNTB
 (5)  006370  062706  000006                      ADD     #6,SP
(12)  006374  013746  002632                      MOV     $GDADR,-(SP)
(11)  006400  013746  002640                      MOV     $BDDAT,-(SP)
(10)  006404  013746  002636                      MOV     $GDDAT,-(SP)
 (9)  006410  012746  004123                      MOV     #TFM1,-(SP)
 (8)  006414  012746  000004                      MOV     #4,-(SP)
 (5)  006420  010600                              MOV     SP,R0
 (6)  006422  104414                              TRAP    C$PNTB
 (6)  006424  062706  000012                      ADD     #12,SP
 (4)  006430                              L10007:
 (4)  006430  104423                              TRAP    C$MSG
4460  006432                                    $MD     6,3,1
 (4)  006432                              ERR6::
 (9)  006432  012746  005764                      MOV     #DH3,-(SP)
 (8)  006436  012746  004114                      MOV     #FM1,-(SP)
 (7)  006442  012746  000002                      MOV     #2,-(SP)
 (4)  006446  010600                              MOV     SP,R0
 (5)  006450  104414                              TRAP    C$PNTB
 (5)  006452  062706  000006                      ADD     #6,SP
(12)  006456  013746  002632                      MOV     $GDADR,-(SP)
(11)  006462  013746  002640                      MOV     $BDDAT,-(SP)
(10)  006466  013746  002636                      MOV     $GDDAT,-(SP)
 (9)  006472  012746  004123                      MOV     #TFM1,-(SP)
 (8)  006476  012746  000004                      MOV     #4,-(SP)
 (5)  006502  010600                              MOV     SP,R0
 (6)  006504  104414                              TRAP    C$PNTB
 (6)  006506  062706  000012                      ADD     #12,SP
 (4)  006512                              L10010:
 (4)  006512  104423                              TRAP    C$MSG
4461  006514                                    $MD     7,3,1
 (4)  006514                              ERR7::
```

```
   (9)  006514  012746  005764          MOV     #DH3,-(SP)
   (8)  006520  012746  004114          MOV     #FM1,-(SP)
   (7)  006524  012746  000002          MOV     #2,-(SP)
   (4)  006530  010600                  MOV     SP,R0
   (5)  006532  104414                  TRAP    C$PNTB
   (5)  006534  062706  000006          ADD     #6,SP
  (12)  006540  013746  002632          MOV     $GDADR,-(SP)
  (11)  006544  013746  002640          MOV     $BDDAT,-(SP)
  (10)  006550  013746  002636          MOV     $GDDAT,-(SP)
   (9)  006554  012746  004123          MOV     #TFM1,-(SP)
   (8)  006560  012746  000004          MOV     #4,-(SP)
   (5)  006564  010600                  MOV     SP,R0
   (6)  006566  104414                  TRAP    C$PNTB
   (6)  006570  062706  000012          ADD     #12,SP
   (4)  006574                  L10011:
   (4)  006574  104423                  TRAP    C$MSG
 4462   006576                          $MD     10,3,1
   (4)  006576                  ERR10::
   (9)  006576  012746  005764          MOV     #DH3,-(SP)
   (8)  006602  012746  004114          MOV     #FM1,-(SP)
   (7)  006606  012746  000002          MOV     #2,-(SP)
   (4)  006612  010600                  MOV     SP,R0
   (5)  006614  104414                  TRAP    C$PNTB
   (5)  006616  062706  000006          ADD     #6,SP
  (12)  006622  013746  002632          MOV     $GDADR,-(SP)
  (11)  006626  013746  002640          MOV     $BDDAT,-(SP)
  (10)  006632  013746  002636          MOV     $GDDAT,-(SP)
   (9)  006636  012746  004123          MOV     #TFM1,-(SP)
   (8)  006642  012746  000004          MOV     #4,-(SP)
   (5)  006646  010600                  MOV     SP,R0
   (6)  006650  104414                  TRAP    C$PNTB
   (6)  006652  062706  000012          ADD     #12,SP
   (4)  006656                  L10012:
   (4)  006656  104423                  TRAP    C$MSG
 4463   006660                          $MD     11,2,2
   (4)  006660                  ERR11::
   (9)  006660  012746  005750          MOV     #DH2,-(SP)
   (8)  006664  012746  004114          MOV     #FM1,-(SP)
   (7)  006670  012746  000002          MOV     #2,-(SP)
   (4)  006674  010600                  MOV     SP,R0
   (5)  006676  104414                  TRAP    C$PNTB
   (5)  006700  062706  000006          ADD     #6,SP
  (11)  006704  013746  002640          MOV     $BDDAT,-(SP)
  (10)  006710  013746  002636          MOV     $GDDAT,-(SP)
   (9)  006714  012746  004146          MOV     #TFM2,-(SP)
   (8)  006720  012746  000003          MOV     #3,-(SP)
   (5)  006724  010600                  MOV     SP,R0
   (6)  006726  104414                  TRAP    C$PNTB
   (6)  006730  062706  000010          ADD     #10,SP
   (4)  006734                  L10013:
   (4)  006734  104423                  TRAP    C$MSG
 4464   006736                          $MD     12,2,2
   (4)  006736                  ERR12::
   (9)  006736  012746  005750          MOV     #DH2,-(SP)
   (8)  006742  012746  004114          MOV     #FM1,-(SP)
   (7)  006746  012746  000002          MOV     #2,-(SP)
```

```
   (4)  006752  010600                        MOV     SP,R0
   (5)  006754  104414                        TRAP    C$PNTB
   (5)  006756  062706  000006                ADD     #6,SP
  (11)  006762  013746  002640                MOV     $BDDAT,-(SP)
  (10)  006766  013746  002636                MOV     $GDDAT,-(SP)
   (9)  006772  012746  004146                MOV     #TFM2,-(SP)
   (8)  006776  012746  000003                MOV     #3,-(SP)
   (5)  007002  010600                        MOV     SP,R0
   (6)  007004  104414                        TRAP    C$PNTB
   (6)  007006  062706  000010                ADD     #10,SP
   (4)  007012                        L10014:
   (4)  007012  104423                        TRAP    C$MSG
 4465   007014                                $MD     13,0,0
   (4)  007014                        ERR13::
   (9)  007014  012746  005716                MOV     #DH0,-(SP)
   (8)  007020  012746  004114                MOV     #TFM1,-(SP)
   (7)  007024  012746  000002                MOV     #2,-(SP)
   (4)  007030  010600                        MOV     SP,R0
   (5)  007032  104414                        TRAP    C$PNTB
   (5)  007034  062706  000006                ADD     #6,SP
   (4)  007040                        L10015:
   (4)  007040  104423                        TRAP    C$MSG
```

```
4467 007042                              $MD     14,2,2
 (4) 007042                      ERR14::
 (9) 007042  012746  005750              MOV     #DH2,-(SP)
 (8) 007046  012746  004114              MOV     #FM1,-(SP)
 (7) 007052  012746  000002              MOV     #2,-(SP)
 (4) 007056  010600                      MOV     SP,R0
 (5) 007060  104414                      TRAP    C$PNTB
 (5) 007062  062706  000006              ADD     #6,SP
(11) 007066  013746  002640              MOV     $BDDAT,-(SP)
(10) 007072  013746  002636              MOV     $GDDAT,-(SP)
 (9) 007076  012746  004146              MOV     #TFM2,-(SP)
 (8) 007102  012746  000003              MOV     #3,-(SP)
 (5) 007106  010600                      MOV     SP,R0
 (6) 007110  104414                      TRAP    C$PNTB
 (6) 007112  062706  000010              ADD     #10,SP
 (4) 007116                      L10016:
 (4) 007116  104423                      TRAP    C$MSG
```

```
4469  007120                                    $MD     15,4,5
 (4)  007120                            ERR15::
 (9)  007120  012746  0060[illegible]4           MOV     #DH4,-(SP)
 (8)  007124  012746  004114                    MOV     #FM1,-(SP)
 (7)  007130  012746  000002                    MOV     #2,-(SP)
 (4)  007134  010600                            MOV     SP,R0
 (5)  007136  104414                            TRAP    C$PNTB
 (5)  007140  062706  000006                    ADD     #6,SP
(11)  007144  013746  002640                    MOV     $BDDAT,-(SP)
(10)  007150  013746  002636                    MOV     $GDDAT,-(SP)
 (9)  007154  012746  004163                    MOV     #TFM5,-(SP)
 (8)  007160  012746  000003                    MOV     #3,-(SP)
 (5)  007164  010600                            MOV     SP,R0
 (6)  007166  104414                            TRAP    C$PNTB
 (6)  007170  062706  000010                    ADD     #10,SP
 (4)  007174                            L10017:
 (4)  007174  104423                            TRAP    C$MSG
4470  007176                                    $MD     16,0,0
 (4)  007176                            ERR16::
 (9)  007176  012746  005716                    MOV     #DH0,-(SP)
 (8)  007202  012746  004114                    MOV     #FM1,-(SP)
 (7)  007206  012746  000002                    MOV     #2,-(SP)
 (4)  007212  010600                            MOV     SP,R0
 (5)  007214  104414                            TRAP    C$PNTB
 (5)  007216  062706  000006                    ADD     #6,SP
 (4)  007222                            L10020:
 (4)  007222  104423                            TRAP    C$MSG
4471  007224                                    $MD     17,0,0
 (4)  007224                            ERR17::
 (9)  007224  012746  005716                    MOV     #DH0,-(SP)
 (8)  007230  012746  004114                    MOV     #FM1,-(SP)
 (7)  007234  012746  000002                    MOV     #2,-(SP)
 (4)  007240  010600                            MOV     SP,R0
 (5)  007242  104414                            TRAP    C$PNTB
 (5)  007244  062706  000006                    ADD     #6,SP
 (4)  007250                            L10021:
 (4)  007250  104423                            TRAP    C$MSG
4472  007252                                    $MD     20,2,2
 (4)  007252                            ERR20::
 (9)  007252  012746  005750                    MOV     #DH2,-(SP)
 (8)  007256  012746  004114                    MOV     #FM1,-(SP)
 (7)  007262  012746  000002                    MOV     #2,-(SP)
 (4)  007266  010600                            MOV     SP,R0
 (5)  007270  104414                            TRAP    C$PNTB
 (5)  007272  062706  000006                    ADD     #6,SP
(11)  007276  013746  002640                    MOV     $BDDAT,-(SP)
(10)  007302  013746  002636                    MOV     $GDDAT,-(SP)
 (9)  007306  012746  004146                    MOV     #TFM2,-(SP)
 (8)  007312  012746  000003                    MOV     #3,-(SP)
 (5)  007316  010600                            MOV     SP,R0
 (6)  007320  104414                            TRAP    C$PNTB
 (6)  007322  062706  000010                    ADD     #10,SP
 (4)  007326                            L10022:
 (4)  007326  104423                            TRAP    C$MSG
4473  007330                                    $MD     21,0,0
 (4)  007330                            ERR21::
```

```
  (9)  007330  012746  005716          MOV     #DH0,-(SP)
  (8)  007334  012746  004114          MOV     #FM1,-(SP)
  (7)  007340  012746  000002          MOV     #2,-(SP)
  (4)  007344  010600                  MOV     SP,R0
  (5)  007346  104414                  TRAP    C$PNTB
  (5)  007350  062706  000006          ADD     #6,SP
  (4)  007354                  L10023:
  (4)  007354  104423                  TRAP    C$MSG
4474   007356                          $MD     22,0,0
  (4)  007356                  ERR22::
  (9)  007356  012746  005716          MOV     #DH0,-(SP)
  (8)  007362  012746  004114          MOV     #FM1,-(SP)
  (7)  007366  012746  000002          MOV     #2,-(SP)
  (4)  007372  010600                  MOV     SP,R0
  (5)  007374  104414                  TRAP    C$PNTB
  (5)  007376  062706  000006          ADD     #6,SP
  (4)  007402                  L10024:
  (4)  007402  104423                  TRAP    C$MSG
4475   007404                          $MD     23,4,5
  (4)  007404                  ERR23::
  (9)  007404  012746  006014          MOV     #DH4,-(SP)
  (8)  007410  012746  004114          MOV     #FM1,-(SP)
  (7)  007414  012746  000002          MOV     #2,-(SP)
  (4)  007420  010600                  MOV     SP,R0
  (5)  007422  104414                  TRAP    C$PNTB
  (5)  007424  062706  000006          ADD     #6,SP
 (11)  007430  013746  002640          MOV     $BDDAT,-(SP)
 (10)  007434  013746  002636          MOV     $GDDAT,-(SP)
  (9)  007440  012746  004163          MOV     #TFM5,-(SP)
  (8)  007444  012746  000003          MOV     #3,-(SP)
  (5)  007450  010600                  MOV     SP,R0
  (6)  007452  104414                  TRAP    C$PNTB
  (6)  007454  062706  000010          ADD     #10,SP
  (4)  007460                  L10025:
  (4)  007460  104423                  TRAP    C$MSG
4476   007462                          $MD     24,0,0
  (4)  007462                  ERR24::
  (9)  007462  012746  005716          MOV     #DH0,-(SP)
  (8)  007466  012746  004114          MOV     #FM1,-(SP)
  (7)  007472  012746  000002          MOV     #2,-(SP)
  (4)  007476  010600                  MOV     SP,R0
  (5)  007500  104414                  TRAP    C$PNTB
  (5)  007502  062706  000006          ADD     #6,SP
  (4)  007506                  L10026:
  (4)  007506  104423                  TRAP    C$MSG
4477   007510                          $MD     25,2,2
  (4)  007510                  ERR25::
  (9)  007510  012746  005750          MOV     #DH2,-(SP)
  (8)  007514  012746  004114          MOV     #FM1,-(SP)
  (7)  007520  012746  000002          MOV     #2,-(SP)
  (4)  007524  010600                  MOV     SP,R0
  (5)  007526  104414                  TRAP    C$PNTB
  (5)  007530  062706  000006          ADD     #6,SP
 (11)  007534  013746  002640          MOV     $BDDAT,-(SP)
 (10)  007540  013746  002636          MOV     $GDDAT,-(SP)
  (9)  007544  012746  004146          MOV     #TFM2,-(SP)
```

```
 (8)  007550  012746  000003          MOV     #3,-(SP)
 (5)  007554  010600                  MOV     SP,R0
 (6)  007556  104414                  TRAP    C$PNTB
 (6)  007560  062706  000010          ADD     #10,SP
 (4)  007564                  L10027:
 (4)  007564  104423                  TRAP    C$MSG
4478  007566                          $MD     26,2,2
 (4)  007566                  ERR26::
 (9)  007566  012746  005750          MOV     #DH2,-(SP)
 (8)  007572  012746  004114          MOV     #FM1,-(SP)
 (7)  007576  012746  000002          MOV     #2,-(SP)
 (4)  007602  010600                  MOV     SP,R0
 (5)  007604  104414                  TRAP    C$PNTB
 (5)  007606  062706  000006          ADD     #6,SP
(11)  007612  013746  002640          MOV     $BDDAT,-(SP)
(10)  007616  013746  002636          MOV     $GDDAT,-(SP)
 (9)  007622  012746  004146          MOV     #TFM2,-(SP)
 (8)  007626  012746  000003          MOV     #3,-(SP)
 (5)  007632  010600                  MOV     SP,R0
 (6)  007634  104414                  TRAP    C$PNTB
 (6)  007636  062706  000010          ADD     #10,SP
 (4)  007642                  L10030:
 (4)  007642  104423                  TRAP    C$MSG
4479  007644                          $MD     27,27,27
 (4)  007644                  ERR27::
 (9)  007644  012746  006030          MOV     #DH27,-(SP)
 (8)  007650  012746  004114          MOV     #FM1,-(SP)
 (7)  007654  012746  000002          MOV     #2,-(SP)
 (4)  007660  010600                  MOV     SP,R0
 (5)  007662  104414                  TRAP    C$PNTB
 (5)  007664  062706  000006          ADD     #6,SP
(12)  007670  013746  002640          MOV     $BDDAT,-(SP)
(11)  007674  013746  002636          MOV     $GDDAT,-(SP)
(10)  007700  013746  002624          MOV     MR0,-(SP)
 (9)  007704  012746  004200          MOV     #TFM27,-(SP)
 (8)  007710  012746  000004          MOV     #4,-(SP)
 (5)  007714  010600                  MOV     SP,R0
 (6)  007716  104414                  TRAP    C$PNTB
 (6)  007720  062706  000012          ADD     #12,SP
 (4)  007724                  L10031:
 (4)  007724  104423                  TRAP    C$MSG
4480  007726                          $MD     28,2,2
 (4)  007726                  ERR28::
 (9)  007726  012746  005750          MOV     #DH2,-(SP)
 (8)  007732  012746  004114          MOV     #FM1,-(SP)
 (7)  007736  012746  000002          MOV     #2,-(SP)
 (4)  007742  010600                  MOV     SP,R0
 (5)  007744  104414                  TRAP    C$PNTB
 (5)  007746  062706  000006          ADD     #6,SP
(11)  007752  013746  002640          MOV     $BDDAT,-(SP)
(10)  007756  013746  002636          MOV     $GDDAT,-(SP)
 (9)  007762  012746  004146          MOV     #TFM2,-(SP)
 (8)  007766  012746  000003          MOV     #3,-(SP)
 (5)  007772  010600                  MOV     SP,R0
 (6)  007774  104414                  TRAP    C$PNTB
 (6)  007776  062706  000010          ADD     #10,SP
```

```
  (4)  010002
  (4)  010002  104423
4481   010004
  (4)  010004
  (9)  010004  012746  006030
  (8)  010010  012746  004114
  (7)  010014  012746  000002
  (4)  010020  010600
  (5)  010022  104414
  (5)  010024  062706  000006
 (12)  010030  013746  002640
 (11)  010034  013746  002636
 (10)  010040  013746  002624
  (9)  010044  012746  004200
  (8)  010050  012746  000004
  (5)  010054  010600
  (6)  010056  104414
  (6)  010060  062706  000012
  (4)  010064
  (4)  010064  104423
4482   010066
  (4)  010066
  (9)  010066  012746  005750
  (8)  010072  012746  004114
  (7)  010076  012746  000002
  (4)  010102  010600
  (5)  010104  104414
  (5)  010106  062706  000006
 (11)  010112  013746  0026~0
 (10)  010116  013746  002636
  (9)  010122  012746  004146
  (8)  010126  012746  000003
  (5)  010132  010600
  (6)  010134  104414
  (6)  010136  062706  000010
  (4)  010142
  (4)  010142  104423
4483   010144
  (4)  010144
  (9)  010144  012746  005716
  (8)  010150  012746  004114
  (7)  010154  012746  000002
  (4)  010160  010600
  (5)  010162  104414
  (5)  010164  062706  000006
  (4)  010170
  (4)  010170  104423
4484   010172
  (4)  010172
  (9)  010172  012746  005716
  (8)  010176  012746  004114
  (7)  010202  012746  000002
  (4)  010206  010600
  (5)  010210  104414
  (5)  010212  062706  000006
  (4)  010216

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

```
  (4)  010216  104423                    TRAP    C$MSG
4485   010220                           $MD     33,2,2
  (4)  010220                    ERR33::
  (9)  010220  012746  005750           MOV     #DH2,-(SP)
  (8)  010224  012746  004114           MOV     #FM1,-(SP)
  (7)  010230  012746  000002           MOV     #2,-(SP)
  (4)  010234  010600                    MOV     SP,R0
  (5)  010236  104414                    TRAP    C$PNTB
  (5)  010240  062706  000006           ADD     #6,SP
 (11)  010244  013746  002640           MOV     $BDDAT,-(SP)
 (10)  010250  013746  002636           MOV     $GDDAT,-(SP)
  (9)  010254  012746  004146           MOV     #TFM2,-(SP)
  (8)  010260  012746  000003           MOV     #3,-(SP)
  (5)  010264  010600                    MOV     SP,R0
  (6)  010266  104414                    TRAP    C$PNTB
  (6)  010270  062706  000010           ADD     #10,SP
  (4)  010274                    L10037:
  (4)  010274  104423                    TRAP    C$MSG
4486   010276                           $MD     34,2,2
  (4)  010276                    ERR34::
  (9)  010276  012746  005750           MOV     #DH2,-(SP)
  (8)  010302  012746  004114           MOV     #FM1,-(SP)
  (7)  010306  012746  000002           MOV     #2,-(SP)
  (4)  010312  010600                    MOV     SP,R0
  (5)  010314  104414                    TRAP    C$PNTB
  (5)  010316  062706  000006           ADD     #6,SP
 (11)  010322  013746  002640           MOV     $BDDAT,-(SP)
 (10)  010326  013746  002636           MOV     $GDDAT,-(SP)
  (9)  010332  012746  004146           MOV     #TFM2,-(SP)
  (8)  010336  012746  000003           MOV     #3,-(SP)
  (5)  010342  010600                    MOV     SP,R0
  (6)  010344  104414                    TRAP    C$PNTB
  (6)  010346  062706  000010           ADD     #10,SP
  (4)  010352                    L10040:
  (4)  010352  104423                    TRAP    C$MSG
4487   010354                           $MD     35,2,2
  (4)  010354                    ERR35::
  (9)  010354  012746  005750           MOV     #DH2,-(SP)
  (8)  010360  012746  004114           MOV     #FM1,-(SP)
  (7)  010364  012746  000002           MOV     #2,-(SP)
  (4)  010370  010600                    MOV     SP,R0
  (5)  010372  104414                    TRAP    C$PNTB
  (5)  010374  062706  000006           ADD     #6,SP
 (11)  010400  013746  002640           MOV     $BDDAT,-(SP)
 (10)  010404  013746  002636           MOV     $GDDAT,-(SP)
  (9)  010410  012746  004146           MOV     #TFM2,-(SP)
  (8)  010414  012746  000003           MOV     #3,-(SP)
  (5)  010420  010600                    MOV     SP,R0
  (6)  010422  104414                    TRAP    C$PNTB
  (6)  010424  062706  000010           ADD     #10,SP
  (4)  010430                    L10041:
  (4)  010430  104423                    TRAP    C$MSG
4488   010432                           $MD     36,2,2
  (4)  010432                    ERR36::
  (9)  010432  012746  005750           MOV     #DH2,-(SP)
  (8)  010436  012746  004114           MOV     #FM1,-(SP)
```

```
   (7)  010442  012746  000002                    MOV     #2,-(SP)
   (4)  010446  010600                             MOV     SP,R0
   (5)  010450  104414                             TRAP    C$PNTB
   (5)  010452  062706  000006                     ADD     #6,SP
   (11) 010456  013746  002640                     MOV     $BDDAT,-(SP)
   (10) 010462  013746  002636                     MOV     $GDDAT,-(SP)
   (9)  010466  012746  004146                     MOV     #TFM2,-(SP)
   (8)  010472  012746  000003                     MOV     #3,-(SP)
   (5)  010476  010600                             MOV     SP,R0
   (6)  010500  104414                             TRAP    C$PNTB
   (6)  010502  062706  000010                     ADD     #10,SP
   (4)  010506                              L10042:
   (4)  010506  104423                             TRAP    C$MSG
 4489
 4490  010510                                      BGNMSG  ERR37
  (3)  010510                              ERR37::
 4491  010510                                      PRINTF  #FM1,#EM1
   (8)  010510  012746  004261                     MOV     #EM1,-(SP)
   (7)  010514  012746  004114                     MOV     #FM1,-(SP)
   (6)  010520  012746  000002                     MOV     #2,-(SP)
   (3)  010524  010600                             MOV     SP,R0
   (4)  010526  104417                             TRAP    C$PNTF
   (4)  010530  062706  000006                     ADD     #6,SP
 4492  010534                                      PRINTF  #TFM37,$GDADR
   (8)  010534  013746  002632                     MOV     $GDADR,-(SP)
   (7)  010540  012746  004225                     MOV     #TFM37,-(SP)
   (6)  010544  012746  000002                     MOV     #2,-(SP)
   (3)  010550  010600                             MOV     SP,R0
   (4)  010552  104417                             TRAP    C$PNTF
   (4)  010554  062706  000006                     ADD     #6,SP
 4493  010560                                      ENDMSG
  (3)  010560                              L10043:
  (3)  010560  104423                             TRAP    C$MSG
 4494
 4495
 4496
```

```
4498                                    .SBTTL  REPORT CODING SECTION
4499
4500
4501                                    ;++
4502                                    ; THE REPORT CODING SECTION CONTAINS THE
4503                                    ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
4504                                    ;--
4505
4506  010562                                    BGNRPT
 (3)  010562                            L$RPT::
4507
4513
4514  010562                                    EXIT    RPT
 (4)  010562  000167                              .WORD   J$JMP
 (3)  010564  000000                              .WORD   L10044-2-.
4515
4522
4523  010566                                    ENDRPT
 (3)  010566                            L10044:
 (3)  010566  104425                              TRAP    C$RPT
4524
4525
4526
4527
4528
```

```
                                        .SBTTL  INITIALIZE SECTION

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

010570                                          BGNINIT
010570                                  L$INIT::

                                        ;INITIALIZE SUBROUTINE STACK
010570  012705  003130                          MOV     #SSTACK,R5
                                        ;STORE BASE LEVEL PROGRAM STACK POINTER
010574  010637  002554                          MOV     SP,PSTACK
010600  005737  002646                          TST     FTIME
010604  001011                                  BNE     1$
010606  013737  000004  002650                  MOV     @#4,SAVE4
010614  013737  000006  002652                  MOV     @#6,SAVE6
010622  012737  000001  002646                  MOV     #1,FTIME
010630  013737  002650  000004  1$:             MOV     SAVE4,@#4
010636  013737  002652  000006                  MOV     SAVE6,@#6
                                        ;SEE IF PROGRAM JUST STARTED, BR IF YES
010644                                          READEF  #EF.START
010644  012700  000040                          MOV     #EF.START,R0
010650  104447                                  TRAP    C$REFG
010652                                          BCOMPLETE       NEWST
010652  103414                                  BCS     NEWST
                                        ;SEE IF THIS IS A NEW PASS, BR IF YES
010654                                          READEF  #EF.NEW
010654  012700  000035                          MOV     #EF.NEW,R0
010660  104447                                  TRAP    C$REFG
010662                                          BCOMPLETE       NEWST
010662  103410                                  BCS     NEWST
                                        ;SEE IF PROGRAM WAS JUST CONTINUED
010664                                          READEF  #EF.CONTINUE
010664  012700  000036                          MOV     #EF.CONTINUE,R0
010670  104447                                  TRAP    C$REFG
010672                                          BCOMPLETE       ENDIT
010672  103570                                  BCS     ENDIT

                                        ;SEE IF PROGRAM JUST RESTARTED, BR IF NOT
010674                                          READEF  #EF.RESTART
010674  012700  000037                          MOV     #EF.RESTART,R0
010700  104447                                  TRAP    C$REFG
010702                                          BNCOMPLETE      GETPRM
010702  103003                                  BCC     GETPRM
010704                                  NEWST:
                                        ;RESET LOGICAL DEVICE TO -1
010704  012737  177777  002552                  MOV     #-1,LOGDEV

                                        ;GET UNIBUS ADRS, VECTOR, PRIORITY LEVEL, LINE UNIT,SWITCH
                                        ;    PACKS, TEST CONNECTOR INFO. FOR THIS M8200,4,7 (CURRENT LOGICAL
                                        ;    DEVICE).
010712                                  GETPRM:
010712  005237  002552                          INC     LOGDEV
010716  023737  002552  002012                  CMP     LOGDEV,L$UNIT
```

```
                                        BGE     NEWST
                                        GPHARD  LOGDEV,R1
                                        MOV     LOGDEV,R0
                                        TRAP    C$GPHRD
                                        MOV     R0,R1
                                        BNCOMPLETF      GETPRM
                                        BCC     GETPRM
                                ;GET ADDRESS OF M8200,4,7
                                        MOV     (R1)+,WTYPE
                                        MOV     (R1),KMCSR
                                ;GET POINTER TO M8200,4,7 CSR HI BYTE
                                        MOV     (R1),KMCSRH
                                        INC     KMCSRH
                                ;GET POINTER TO M8200,4,7 CTL OUT REG
                                        MOV     (R1),KMCTL
                                        ADD     #2,KMCTL
                                ;GET POINTER TO M8200,4,7 PORT REG - SEL 4
                                        MOV     (R1),KMP04
                                        ADD     #4,KMP04
                                ;GET POINTER TO M8200,4,7 PORT REG - SEL 6
                                        MOV     (R1)+,KMP06
                                        ADD     #6,KMP06
                                ;GET POINTER TO RCV VECTOR
                                        MOV     (R1),KMRVEC
                                ;GET POINTER TO RCV PRIORITY LEVEL
                                        MOV     (R1),KMRLVL
                                        ADD     #2,KMRLVL
                                ;GET POINTER TO TX VECTOR
                                        MOV     (R1),KMTVEC
                                        ADD     #4,KMTVEC
                                ;GET POINTER TO TX PRIORITY LEVEL
                                        MOV     (R1),KMTLVL
                                        ADD     #6,KMTLVL
                                ;PUT VECTOR INTO STAT1
                                        MOV     (R1)+,STAT1
                                ;PUT PRIORITY INTO STAT1
                                        BIS     (R1)+,STAT1
                                ;SEE IF NO LINE UNIT, SET BIT IF YES
                                        TST     (R1)
                                        BNE     50000$
                                        BIS     #BIT12,STAT1
                                        BR      4$
                                50000$:
                                ;SEE IF M8201 LINE UNIT, SET BIT IF YES
                                        CMP     (R1),#1
                                        BNE     50001$
                                        BR      4$
                                50001$:
                                ;SEE IF M8202 LINE UNIT, SET BIT IF YES
                                        CMP     (R1),#2
                                        BNE     50002$
                                        BIS     #BIT13,STAT1
                                        BR      4$
                                50002$:
                                ;SET BIT FOR M8203 LINE UNIT
                                        BIS     #BIT15,STAT1
```

```
4625  011140                                  4$:
4626                                          ;SET BIT IN STAT1 FOR TEST CONNECTOR
4627  011140  056137  000006  002700          BIS     6(R1),STAT1
4628  011146  062701  000002                  ADD     #2,R1
4629                                          ;SET SWITCH PACK #1 IN STAT2 LOW BYTE
4630  011152  012137  002702                  MOV     (R1)+,STAT2
4631                                          ;SET SWITCH PACK #2 IN STAT2 HIGH BYTE
4632  011156  111137  002703                  MOVB    (R1),STAT2+1
4633
4634                                          ;INCREMENT LOGICAL UNIT (DEVICE) NUMBER
4635                                          ;       INC     LOGDEV
4636  011162  000240                          NOP
4637  011164  000240                          NOP
4638
4639  011166  012737  002000  002606          MOV     #2000,MEMSZ
4640  011174  005037  002630                  CLR     TYPE
4641  011200  123727  002626  000000          CMPB    WTYPE,#0
4642  011206  001422                          BEQ     ENDIT
4643  011210  123727  002626  000004          CMPB    WTYPE,#4        ;KMC?
4644  011216  001004                          BNE     5$
4645  011220  012737  000001  002630          MOV     #1,TYPE
4646  011226  000412                          BR      ENDIT
4647  011230  012737  003777  002606  5$:     MOV     #3777,MEMSZ
4648  011236  123727  002626  000006          CMPB    WTYPE,#6
4649  011244  001003                          BNE     ENDIT
4650  011246  012737  000001  002630          MOV     #1,TYPE
4651  011254                          ENDIT:
4652  011254                                  ENDINIT
 (3)  011254                          L10045:
 (3)  011254  104411                          TRAP    C$INIT
4653
4654                                  .EVEN
4655  011256                                  BGNAUTO
 (3)  011256                          L$AUTO::
4656                                          ;DEVICE DOES NOT HAVE A 'READY'
4657  011256  013701  002716                  MOV     KMCSR,R1        ;R1 CONTAINS BASE M8200,4,7 ADDRESS
4658  011262  012705  000004                  MOV     #4,R5           ;4 REGISTERS TO BE TESTED
4659  011266  012737  011320  000004          MOV     #2$,4           ;SET UP TIMEOUT TRAP
4660  011274  012737  000340  000006          MOV     #340,6          ;LEVEL 7
4661  011302  005711                  1$:     TST     (R1)            ;REFERENCE DEVICE REGISTER
4662  011304  000240                          NOP
4663  011306  062701  000002                  ADD     #2,R1           ;NEXT REGISTER
4664  011312  005305                          DEC     R5              ;DEC REGISTER COUNT
4665  011314  001372                          BNE     1$              ;BR IF NOT LAST REGISTER
4666  011316  000407                          BR      3$
4667  011320  062706  000004          2$:     ADD     #4,SP
4668  011324  010137  002632                  MOV     R1,$GDADR
4669  011330                                  DODJ    LOGDEV
 (3)  011330  013700  002552                  MOV     LOGDEV,R0
 (3)  011334  104451                          TRAP    C$DODU
4670
4671  011336  013737  002650  000004  3$:     MOV     SAVE4,4
4672  011344  013737  002652  000006          MOV     SAVE6,6
4673  011352                                  ENDAUTO
 (3)  011352                          L10046:
 (3)  011352  104461                          TRAP    C$AUTO
```

4674

```
                        .SBTTL  CLEANUP CODING SECTION

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

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

011356                          ENDCLN
011356                  L$0047:
011356  104412                  TRAP    C$CLEAN
```

```
                                .SBTTL  DROP UNIT SECTION

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

011360                                  BGNDU
(3) 011360                      L$DU::
                                ;ISSUE UNIBUS RESET TO CLEAN UP
011360                                  BRESET
(3) 011360  104433                      TRAP    C$RESET
011362                                  ENDDU
(3) 011362                      L10050:
(3) 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
                        BADHEAD
                        ;********************* TEST 1 ** *****************
                        ;*VERIFY THAT REFERENCING UNIBUS DEVICE REGISTERS
                        ;*DOES NOT CAUSE A TIME OUT TRAP
                        BADHEAD
                        ;********************* TEST 1 ********************
                        BGNTST
                        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
                        T2::
011514                          MYINT
011514 013701 002716            MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
011520 005011                   CLR     (R1)            ;CLEAR KMCSR
011522 005002                   CLR     R2              ;CLEAR 'EXPECTED''
```

```
 4766  011524  011104                        MOV     (R1),R4         ;PUT KMCSR IN 'FOUND'
 4767  011526  001413                        BEQ     1$              ;BR IF CLEARED
 4768  011530                                ERROR   26              ;ERROR KMCSR NOT CLEARED
  (5)  011546  104455                        TRAP    C$ERDF
  (6)  011550  000032                        .WORD   26
  (6)  011552  005474                        .WORD   EM26
  (6)  011554  007566                        .WORD   ERR26
 4769  011556                          1$:
 4770  011556                          ENDTST
  (3)  011556                          L10053:
  (3)  011556  104401                        TRAP    C$ETST
 4771
 4772  011560                                BADHEAD
  (2)                                       ;******************** TEST 3 ********************
 4773                                       ;*UNIBUS REGISTER WORD DUAL ADDRESSING TEST
 4774                                       ;*LOAD ALL REGISTERS WITH INCREMENTING PATTERN
 4775                                       ;*READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING
 4776                                       ;*THE SEQUENCE:
 4777                                       ;*      1. CLEAR REGISTER
 4778                                       ;*      2. WRITE PATTERN
 4779                                       ;*      3. VERIFY PATTERN
 4780                                       ;*      4. DO ALL 4 REGISTERS
 4781                                       ;*      5. READ ALL BACK IF ERRORS,
 4782                                       ;*         DUAL ADDRESS PROBLEM.
 4783                                       ;*
 4784                                       ;*            1 IN REG 0
 4785                                       ;*            2 IN REG 2
 4786                                       ;*            3 IN REG 4
 4787                                       ;*            4 IN REG 6
 4788  011560                                BADHEAD
  (2)                                       ;******************** TEST 3 ********************
 4789
 4790  011560                          BGNTST
  (3)  011560                          T3::
 4791  011560                                MYINT
  (1)  011560  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 4792                                                                  ;R1 CONTAINS BASE M8200,4,7 ADDRESS
 4793  011564                                MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  011564  004537  003156                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 4794  011570  012702  000001                MOV     #1,R2           ;START PATTERN AT 1
 4795  011574                                BGNSEG
  (3)  011574  104404                        TRAP    C$BSEG
 4796  011576  005011                  1$:   CLR     (R1)            ;CLEAR REGISTER
 4797  011600  010211                        MOV     R2,(R1)         ;WRITE M8200,4,7 REGISTER WITH PATTERN
 4798  011602  011104                        MOV     (R1),R4         ;READ M8200,4,7 REGISTER INTO 'FOUND'
 4799  011604  020204                        CMP     R2,R4           ;IS DATA CORRECT
 4800  011606  001413                        BEQ     2$              ;BR IS YES
 4801  011610                                ERROR   26              ;DATA ERROR
  (5)  011626  104455                        TRAP    C$ERDF
  (6)  011630  000032                        .WORD   26
  (6)  011632  005474                        .WORD   EM26
  (6)  011634  007566                        .WORD   ERR26
 4802  011636                          2$:   ESCAPE  SEG
  (3)  011636  104410                        TRAP    C$ESCAPE
  (3)  011640  000014                        .WORD   10000$-.
 4803  011642  005721                        TST     (R1)+           ;NEXT REGISTER
```

```
 4804  011644  005202                        INC     R2              ;INCREMENT DATA PATTERN
 4805  011646  022702  000005                CMP     #5,R2           ;LAST REGISTER?
 4806  011652  001351                        BNE     1$              ;BR IF NO
 4807  011654                                ENDSEG
   (3) 011654                          10000$:
   (3) 011654  104405                        TRAP    C$ESEG
 4808  011656  013701  002716                MOV     KMCSR,R1        ;BASE 18200,4,7 ADDRESS TO R1
 4809  011662  012702  000001                MOV     #1,R2           ;RESTART PATTERN AT 1
 4810  011666                                BGNSEG
   (3) 011666  104404                        TRAP    C$BSEG
 4811  011670                              3$:
 4812  011670  011104                        MOV     (R1),R4         ;READ COMM. MICR-PROCESSOR FAMILY REGISTER INTO 'FOUND'
 4813  011672  020204                        CMP     R2,R4           ;IS DATA CORRECT
 4814  011674  001413                        BEQ     4$              ;BR IF YES
 4815  011676                                ERROR   2               ;DUAL ADDRESSING ERROR
   (5) 011714  104455                        TRAP    C$ERDF
   (6) 011716  000002                        .WORD   2
   (6) 011720  004307                        .WORD   EM2
   (6) 011722  006132                        .WORD   ERR2
 4816  011724                              4$:     ESCAPE  SEG
   (3) 011724  104410                        TRAP    C$ESCAPE
   (3) 011726  000014                        .WORD   10001$-.
 4817  011730  005721                        TST     (R1)+           ;NEXT REGISTER
 4818  011732  005202                        INC     R2              ;INCREMENT PATTERN
 4819  011734  022702  000005                CMP     #5,R2           ;LAST REGISTER?
 4820  011740  001353                        BNE     3$              ;BR IF NO
 4821  011742                                ENDSEG
   (3) 011742                          10001$:
   (3) 011742  104405                        TRAP    C$ESEG
 4822  011744                              ENDTST
   (3) 011744                          L10054:
   (3) 011744  104401                        TRAP    C$ETST
 4823
 4824  011746                                BADHEAD
   (2)                                       ;********************* TEST 4 *********************
 4825                                        ;*CONTROL STATUS REGISTER WRITE/READ TEST
 4826                                        ;*FLOAT A ONE THROUGH BSEL 0
 4827                                        ;*CLEAR BIT0, VERIFY BIT0 WAS CLEARED
 4828  011746                                BADHEAD
   (2)                                       ;********************* TEST 4 *********************
 4829
 4830  011746                              BGNTST
   (3) 011746                          T4::
 4831  011746                                MSTCLR                  ;MASTER CLEAR M8200,4,7
   (1) 011746  004537  003156                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 4832  011752  005037  002624                CLR     MR0
 4833  011756  012702  000001                MOV     #BIT0,R2        ;INDICATE BSEL0
 4834  011762                                BGNSEG
   (3) 011762  104404                        TRAP    C$BSEG
 4835  011764  013701  002716              1$:     MOV     KMCSR,R1        ;PUT REGISTER ADDRESS IN R1
 4836  011770  010237  002636                MOV     R2,$GDDAT
 4837  011774  013711  002636                MOV     $GDDAT,(R1)             ;WRITE BIT 0
 4838  012000  011104                        MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
 4839  012002  023704  002636                CMP     $GDDAT,R4               ;IS DATA CORRECT
 4840  012006  001411                        BEQ     2$              ;BR IF YES
 4841  012010                                ERROR   27,YES          ;DATA ERROR
```

```
   (5)  012022  104455                          TRAP    C$ERDF
   (6)  012024  000033                          .WORD   27
   (6)  012026  005525                          .WORD   EM27
   (6)  012030  007644                          .WORD   ERR27
 4842   012032                          2$:     ESCAPE  SEG
   (3)  012032  104410                          TRAP    C$ESCAPE
   (3)  012034  000052                          .WORD   10000$-.
 4843   012036  040211                  3$:     BIC     R2,(R1)         ;CLEAR BSEL0
 4844   012040  005037  002636                  CLR     $GDDAT          ;CLEAR 'EXPECTED'
 4845   012044  011104                          MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
 4846   012046  001413                          BEQ     4$              ;BR IF ZERO
 4847   012050                                  ERROR   2               ;DATA ERROR BSEL NOT CLEARED
   (5)  012066  104455                          TRAP    C$ERDF
   (6)  012070  000002                          .WORD   2
   (6)  012072  004307                          .WORD   EM2
   (6)  012074  006132                          .WORD   ERR2
 4848   012076                          4$:     ESCAPE  SEG
   (3)  012076  104410                          TRAP    C$ESCAPE
   (3)  012100  000006                          .WORD   10000$-.
 4849   012102  106302                          ASLB    R2
 4850   012104  001327                          BNE     1$
 4851   012106                                  ENDSEG
   (3)  012106                          '0000$:
   (3)  012106  104405                          TRAP    C$ESEG
 4852   012110                          ENDTST
   (3)  012110                          L10055:
   (3)  012110  104401                          TRAP    C$ETST
 4853
 4854
```

```
012112                                      BADHEAD
                                            ;******************* TEST 5 ** *****************
                                            ;*CONTROL STATUS REGISTER WRITE/READ TEST
                                            ;*SET BIT9, VERIFY BIT9 WAS SET
                                            ;*CLEAR BIT9, VERIFY BIT9 WAS CLEARED
012112                                      BADHEAD
                                            ;******************* TEST 5 *******************

012112                              BGNTST
012112                              T5::
012112                                      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
012156  104455                              TRAP    C$ERDF
012160  000032                              .WORD   26
012162  005474                              .WORD   EM26
012164  007566                              .WORD   ERR26
012166                              2$:     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                      ;CLEAP '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:
```

```
  (3)  012244  104401                    TRAP    C$ETST
 4889
 4890  012246                           BADHEAD
  (2)                                   ;****************** TEST 6 ******************
 4891                                   ;*CONTROL STATUS REGISTER WRITE/READ TEST
 4892                                   ;*SET BIT11, VERIFY BIT11 WAS SET
 4893                                   ;*CLEAR BIT11, VERIFY BIT11 WAS CLEARED
 4894  012246                           BADHEAD
  (2)                                   ;****************** TEST 6 ******************
 4895
 4896  012246                    BGNTST
  (3)  012246                    T6::
 4897  012246                           MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  012246  004537  003156           JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 4898  012252                           BGNSEG
  (3)  012252  104404                   TRAP    C$BSEG
 4899  012254  013701  002716    1$:    MOV     KMCSR,R1        ;PUT REGISTER ADDRESS IN R1
 4900  012260  012702  004000           MOV     #BIT11,R2       ;PUT DATA IN 'EXPECTED'
 4901  012264  010211                   MOV     R2,(R1)         ;WRITE BIT 11
 4902  012266  011104                   MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
 4903  012270  020204                   CMP     R2,R4           ;IS DATA CORRECT
 4904  012272  001413                   BEQ     2$              ;BR IF YES
 4905  012274                           ERROR   26              ;DATA ERROR
  (5)  012312  104455                   TRAP    C$ERDF
  (6)  012314  000032                   .WORD   26
  (6)  012316  005474                   .WORD   EM26
  (6)  012320  007566                   .WORD   ERR26
 4906  012322                    2$:    ESCAPE  SEG
  (3)  012322  104410                   TRAP    C$ESCAPE
  (3)  012324  000002                   .WORD   10000$-.
 4907  012326                           ENDSEG
  (3)  012326                    10000$:
  (3)  012326  104405                   TRAP    C$ESEG
 4908  012330                           BGNSEG
  (3)  012330  104404                   TRAP    C$BSEG
 4909  012332  042711  004000    3$:    BIC     #BIT11,(R1)     ;CLEAR BIT 11
 4910  012336  005002                   CLR     R2              ;CLEAR 'EXPECTED'
 4911  012340  011104                   MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
 4912  012342  001414                   BEQ     4$              ;BR IF ZERO
 4913  012344                           ERROR   26              ;DATA ERROR BIT11 NOT CLEARED
  (5)  012362  104455                   TRAP    C$ERDF
  (6)  012364  000032                   .WORD   26
  (6)  012366  005474                   .WORD   EM26
  (6)  012370  007566                   .WORD   ERR26
 4914  012372                           ENDSEG
  (3)  012372                    10001$:
  (3)  012372  104405                   TRAP    C$ESEG
 4915  012374                    4$:
 4916  012374                    ENDTST
  (3)  012374                    L10057:
  (3)  012374  104401                   TRAP    C$ETST
 4917
 4918  012376                           BADHEAD
  (2)                                   ;****************** TEST 7 ******************
 4919                                   ;*CONTROL STATUS REGISTER WRITE/READ TEST
 4920                                   ;*SET BIT12, VERIFY BIT12 WAS SET
```

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

                                BGNTST
                                T7::
                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
012376  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
                                        BGNSEG
012402  104404                          TRAP    C$BSEG
012404  013701  002716          1$:     MOV     KMCSR,R1        ;PUT REGISTER ADDRESS IN R1
012410  012702  010000                  MOV     #BIT12,R2       ;PUT DATA IN 'EXPECTED'
012414  010211                          MOV     R2,(R1)         ;WRITE BIT 12
012416  011104                          MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
012420  020204                          CMP     R2,R4           ;IS DATA CORRECT
012422  001413                          BEQ     2$              ;BR IF YES
012424                                  ERROR   26              ;DATA ERROR
012442  104455                          TRAP    C$ERDF
012444  000032                          .WORD   26
012446  005474                          .WORD   EM26
012450  007566                          .WORD   ERR26
012452                          2$:     ESCAPE  SEG
012452  104410                          TRAP    C$ESCAPE
012454  000002                          .WORD   10000$-.
012456                                  ENDSEG
                                10000$:
012456  104405                          TRAP    C$ESEG
012460                                  BGNSEG
012460  104404                          TRAP    C$BSEG
012462  042711  010000          3$:     BIC     #BIT12,(R1)     ;CLEAR BIT 12
012466  005002                          CLR     R2              ;CLEAR 'EXPECTED'
012470  011104                          MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
012472  001414                          BEQ     4$              ;BR IF ZERO
012474                                  ERROR   26              ;DATA ERROR BIT12 NOT CLEARED
012512  104455                          TRAP    C$ERDF
012514  000032                          .WORD   26
012516  005474                          .WORD   EM26
012520  007566                          .WORD   ERR26
012522                                  ENDSEG
                                10001$:
012522  104405                          TRAP    C$ESEG
012524                          4$:
                                ENDTST
012524                          L10060:
012524  104401                          TRAP    C$ETST

012526                                  BADHEAD
                                        ;****************** TEST 8 ******************
                                        ;*CONTROL OUT REGISTER WRITE/READ TEST
                                        ;*FLOAT A ONE THROUGH SEL2
012526                                  BADHEAD
                                        ;****************** TEST 8 ******************

012526                          BGNTST
012526                          T8::
012526                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
```

```
  (1)  012526  004537  003156                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
4953   012532  012737  000002  002624        MOV     #2,MR0
4954   012540  012702  000001                MOV     #1,R2
4955   012544                                BGNSEG
  (3)  012544  104404                        TRAP    C$BSEG
4956
4957   012546  013701  002722          1$:   MOV     KMCTL,R1          ;PUT REGISTER ADDRESS IN R1
4958   012552  010237  002636                MOV     R2,$GDDAT         ;PUT DATA IN 'EXPECTED'
4959   012556  013711  002636                MOV     $GDDAT,(R1)               ;WRITE BIT 0
4960   012562  011104                        MOV     (R1),R4           ;READ CONTROL OUT REGISTER
4961   012564  023704  002636                CMP     $GDDAT,R4                 ;IS DATA CORRECT
4962   012570  001411                        BEQ     2$                ;BR IF YES
4963   012572                                ERROR   27,YES            ;DATA ERROR
  (5)  012604  104455                        TRAP    C$ERDF
  (6)  012606  000033                        .WORD   27
  (6)  012610  005525                        .WORD   EM27
  (6)  012612  007644                        .WORD   ERR27
4964   012614                          2$:   ESCAPE  SEG
  (3)  012614  104410                        TRAP    C$ESCAPE
  (3)  012616  000046                        .WORD   10000$-.
4965   012620  040211                  3$:   BIC     R2,(R1) ;CLEAR BIT
4966   012622  005037  002636                CLR     $GDDAT            ;CLEAR 'EXPECTED'
4967   012626  011104                        MOV     (R1),R4           ;READ CONTROL OUT REGISTER
4968   012630  001411                        BEQ     4$                ;BR IF ZERO
4969   012632                                ERROR   27,YES            ;DATA ERROR BIT0 NOT CLEARED
  (5)  012644  104455                        TRAP    C$ERDF
  (6)  012646  000033                        .WORD   27
  (6)  012650  005525                        .WORD   EM27
  (6)  012652  007644                        .WORD   ERR27
4970   012654                          4$:   ESCAPEE SEG
  (3)  012654  104410                        TRAP    C$ESCAPE
  (3)  012656  000006                        .WORD   10000$-.
4971   012660  006302                        ASL     R2
4972   012662  001331                        BNE     1$
4973   012664                                ENDSEG
  (3)  012664                     10000$:
  (3)  012664  104405                        TRAP    C$ESEG
4974   012666                          ENDTST
  (3)  012666                           L10061:
  (3)  012666  104401                        TRAP    C$ETST
4975
4976
4977
4978
4979
4980
4981   012670                                BADHEAD
  (2)                                         ;********************* TEST 9 *********************
4982                                          ;*PORT4 REGISTER WRITE/READ TEST
4983                                          ;*FLOAT A ONE THROUGH PORT4 REGISTER
4984                                          ;*FLOAT A ZERO THROUGH PORT4 REGISTER
4985   012670                                BADHEAD
  (2)                                         ;********************* TEST 9 *********************
4986
4987
4988   012670                          BGNTST
```

```
 (3)  012670                                T9::
      012670  012737  000004  002624                MOV     #4,MR0
      012676                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  012676  004537  003156                        JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
      012702  013701  002724                        MOV     KMP04,R1        ;PUT REGISTER ADDRESS IN R1
      012706  012702  000001                        MOV     #1,R2            ;START WITH BIT0
      012712                                        BGNSEG
 (3)  012712  104404                                TRAP    C$BSEG
      012714                                64$:
      012714  010211                                MOV     R2,(R1)         ;WRITE PORT4 REGISTER
      012716  011104                                MOV     (R1),R4         ;READ PORT4 REGISTER
      012720  020204                                CMP     R2,R4           ;COMPARE EXPECTED AND FOUND
      012722  001413                                BEQ     65$             ;BR IF OK
      012724                                        ERROR   27              ;WRITE/READ ERROR
 (5)  012742  104455                                TRAP    C$ERDF
 (6)  012744  000033                                .WORD   27
 (6)  012746  005525                                .WORD   EM27
 (6)  012750  007644                                .WORD   ERR27
      012752                                65$:    ESCAPE  SEG
 (3)  012752  104410                                TRAP    C$ESCAPE
 (3)  012754  000010                                .WORD   10000$-.
      012756  000241                                CLC                     ;CLEAR CARRY
      012760  006102                                ROL     R2              ;SHIFT TO NEXT BIT
      012762  001354                                BNE     64$             ;BR IF NOT DONE YET?
      012764                                        ENDSEG
 (3)  012764                                10000$:
 (3)  012764  104405                                TRAP    C$ESEG
      012766  012702  000001                        MOV     #1,R2           ;START WITH BIT0
      012772                                        BGNSEG
 (3)  012772  104404                                TRAP    C$BSEG
      012774                                66$:
      012774  005102                                COM     R2              ;CHANGE TO A FLOATING ZERO
      012776  010211                                MOV     R2,(R1)         ;WRITE PORT4 REGISTER
      013000  011104                                MOV     (R1),R4
      013002  020204                                CMP     R2,R4           ;COMPARE EXPECTED AND FOUND
      013004  001413                                BEQ     67$             ;BR IF OK
      013006                                        ERROR   27              ;WRITE/READ ERROR
 (5)  013024  104455                                TRAP    C$ERDF
 (6)  013026  000033                                .WORD   27
 (6)  013030  005525                                .WORD   EM27
 (6)  013032  007644                                .WORD   ERR27
      013034                                67$:    ESCAPE  SEG
 (3)  013034  104410                                TRAP    C$ESCAPE
 (3)  013036  000012                                .WORD   10001$-.
      013040  005102                                COM     R2              ;CHANGE BACK TO A FLOATING ONE
      013042  000241                                CLC                     ;CLEAR CARRY
      013044  006102                                ROL     R2              ;SHIFT TO NEXT BIT
      013046  001352                                BNE     66$             ;BR IF NOT DONE YET?
      013050                                        ENDSEG
 (3)  013050                                10001$:
 (3)  013050  104405                                TRAP    C$ESEG
      013052                                ENDTST
 (3)  013052                                L10062:
 (3)  013052  104401                                TRAP    C$ETST

      013054                                        BADHEAD
```

```
 (2)                                    ;****************** TEST 10 ******************
5023                                    ;*PORT6 REGISTER WRITE/READ TEST
5024                                    ;*FLOAT A ONE THROUGH PORT6 REGISTER
5025                                    ;*FLOAT A ZERO THROUGH PORT6 REGISTER
5026  013054                            BADHEAD
 (2)                                    ;****************** TEST 10 ******************
5027
5028  013054                      BGNTST
 (3)  013054                      T10::
5029  013054  012737  000006  002624        MOV     #6,MR0
5030  013062                                 MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  013062  004537  003156                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
5031  013066  013701  002726                MOV     KMP06,R1        ;PUT REGISTER ADDRESS IN R1
5032  013072  012702  000001                MOV     #1,R2            ;START WITH BIT0
5033  013076                                 BGNSEG
 (3)  013076  104404                       TRAP    C$BSEG
5034  013100                      64$:
5035  013100  010211                        MOV     R2,(R1)         ;WRITE PORT6 REGISTER
5036  013102  011104                        MOV     (R1),R4         ;READ PORT6 REGISTER
5037  013104  020204                        CMP     R2,R4           ;COMPARE EXPECTED AND FOUND
5038  013106  001413                        BEQ     65$             ;BR IF OK
5039  013110                                 ERROR   27              ;WRITE/READ ERROR
 (5)  013126  104455                       TRAP    C$ERDF
 (6)  013130  000033                       .WORD   27
 (6)  013132  005525                       .WORD   EM27
 (6)  013134  007644                       .WORD   ERR27
5040  013136                      65$:      ESCAPE  SEG
 (3)  013136  104410                       TRAP    C$ESCAPE
 (3)  013140  000010                       .WORD   10000$-.
5041  013142  000241                        CLC                     ;CLEAR CARRY
5042  013144  006105                        ROL     R5              ;SHIFT TO NEXT BIT
5043  013146  001354                        BNE     64$             ;BR IF NOT DONE YET?
5044  013150                                 ENDSEG
 (3)  013150                      10000$:
 (3)  013150  104405                       TRAP    C$ESEG
5045  013152  012702  000001                MOV     #1,R2           ;START WITH BIT0
5046  013156                                 BGNSEG
 (3)  013156  104404                       TRAP    C$BSEG
5047  013160                      66$:
5048  013160  005102                        COM     R2              ;CHANGE TO A FLOATING ZERO
5049
5050  013162  010211                        MOV     R2,(R1)         ;WRITE PORT6 REGISTER
5051  013164  011104                        MOV     (R1),R4         ;READ PORT6 REGISTER
5052  013166  020204                        CMP     R2,R4           ;COMPARE EXPECTED AND FOUND
5053  013170  001413                        BEQ     67$             ;BR IF OK
5054  013172                                 ERROR   27              ;WRITE/READ ERROR
 (5)  013210  104455                       TRAP    C$ERDF
 (6)  013212  000033                       .WORD   27
 (6)  013214  005525                       .WORD   EM27
 (6)  013216  007644                       .WORD   ERR27
5055  013220                      67$:      ESCAPE  SEG
 (3)  013220  104410                       TRAP    C$ESCAPE
 (3)  013222  000012                       .WORD   10001$-.
5056  013224  005102                        COM     R2              ;CHANGE BACK TO A FLOATING ONE
5057  013226  000241                        CLC                     ;CLEAR CARRY
5058  013230  006102                        ROL     R2              ;SHIFT TO NEXT BIT
```

```
5059  013232  001352                    BNE     66$             ;BR IF NOT DONE YET?
5060  013234                            ENDSEG
 (3)  013234                    10001$:
 (3)  013234  104405                    TRAP    C$ESEG
5061  013236                    ENDTST
 (3)  013236                    L10063:
 (3)  013236  104401                    TRAP    C$ETST
5062
5063  013240                            BADHEAD
 (2)                                    ;****************** TEST 11 ******************
5064                                    ;*UNIBUS REGISTER BYTE DUAL ADDRESSING TEST
5065                                    ;*LOAD ALL REGISTERS WITH INCREMENTING PATTERN
5066                                    ;*READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING
5067  013240                            BADHEAD
 (2)                                    ;****************** TEST 11 ******************
5068
5069  013240                    BGNTST
 (3)  013240                    T11::
5070  013240                            MYINT
 (1)  013240  013701  002716            MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5071  013244                            MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  013244  004537  003156            JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
5072  013250  012702  000001            MOV     #1,R2           ;START PATTERN AT 1
5073  013254                            BGNSEG
 (3)  013254  104404                    TRAP    C$BSEG
5074  013256  105011            1$:     CLRB    (R1)            ;CLEAR REGISTER
5075  013260  110211                    MOVB    R2,(R1)         ;WRITE M8200,4,7 REGISTER WITH PATTERN
5076  013262  111104                    MOVB    (R1),R4         ;READ M8200,4,7 REGISTER INTO 'FOUND'
5077  013264  120204                    CMPB    R2,R4           ;IS DATA CORRECT
5078  013266  001413                    BEQ     2$              ;BR IF YES
5079  013270                            ERROR   2               ;DATA ERROR
 (5)  013306  104455                    TRAP    C$ERDF
 (6)  013310  000002                    .WORD   2
 (6)  013312  004307                    .WORD   EM2
 (6)  013314  006132                    .WORD   ERR2
5080  013316                    2$:     ESCAPE  SEG
 (3)  013316  104410                    TRAP    C$ESCAPE
 (3)  013320  000024                    .WORD   10000$-.
5081  013322  105721                    TSTB    (R1)+           ;NEXT REGISTER
5082  013324  005202                    INC     R2              ;INCREMENT DATA PATTERN
5083  013326  022702  000011            CMP     #11,R2          ;LAST REGISTER?
5084  013332  001351                    BNE     1$              ;BR IF NO
5085  013334  013701  002716            MOV     KMCSR,R1        ;BASE M8200,4,7 ADDRESS TO R1
5086  013340  012702  000001            MOV     #1,R2           ;RESTART PATTERN AT 1
5087  013344                            ENDSEG
 (3)  013344                    10000$:
 (3)  013344  104405                    TRAP    C$ESEG
5088  013346                            BGNSEG
 (3)  013346  104404                    TRAP    C$BSEG
5089  013350                    3$:
5090  013350  111104                    MOVB    (R1),R4         ;READ COMM.MICRO-PROCESSOR FAMILY REGISTER INTO 'FOUND'
5091  013352  120204                    CMPB    R2,R4           ;IS DATA CORRECT
5092  013354  001413                    BEQ     4$              ;BR IF YES
5093  013356                            ERROR   2               ;DUAL ADDRESSING ERROR
 (5)  013374  104455                    TRAP    C$ERDF
 (6)  013376  000002                    .WORD   2
```

```
 (6)  013400  004307                        .WORD   EM2
 (6)  013402  006132                        .WORD   ERR2
5094  013404                        4$:     ESCAPE  SEG
 (3)  013404  104410                        TRAP    C$ESCAPE
 (3)  013406  000014                        .WORD   1000*$-.
5095  013410  105721                        TSTB    (R1)+           ;NEXT REGISTER
5096  013412  005202                        INC     R2              ;INCREMENT PATTERN
5097  013414  022702  000011                CMP     #11,R2          ;LAST REGISTER?
5098  013420  001353                        BNE     3$              ;BR IF NO
5099  013422                                ENDSEG
 (3)  013422                        10001$:
 (3)  013422  104405                        TRAP    C$ESEG
5100  013424                        ENDTST
 (3)  013424                        L10064:
 (3)  013424  104401                        TRAP    C$ETST
5101
5102  013426                                BADHEAD
 (2)                                        ;****************** TEST 12 ******************
5103                                        ;*MAINTENANCE INSTRUCTION REGISTER TEST
5104                                        ;*VERIFY THAT THE MAINT IR CAN BE WRITTEN TO ALL ZEROS'
5105                                        ;*AND ALL ONES'. VERIFY THAT IT IS CLEARED ON A BUS RESET.
5106  013426                                BADHEAD
 (2)                                        ;****************** TEST 12 ******************
5107
5108  013426                        BGNTST
 (3)  013426                        112::
5109                                                        ;R1 CONTAINS BASE M8200,4,7 ADDRESS
5110  013426                                MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  013426  004537  003156                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
5111  013432                                MYINT
 (1)  013432  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5112  013436                                BGNSEG
 (3)  013436  104404                        TRAP    C$BSEG
5113  013440  012711  003000                MOV     #BIT9!BIT10,(R1) ;SEL6 IS NOW THE IR
5114  013444  005002                        CLR     R2              ;PUT 'EXPECTED'' IN $GDDAT
5115  013446  010261  000006        1$:     MOV     R2,6(R1)        ;CLEAR THE IR
5116  013452  016104  000006                MOV     6(R1),R4        ;READ THE IR
5117  013456  020204                        CMP     R2,R4           ;IS IT CLEARED?
5118  013460  001413                        BEQ     2$              ;BR IF YES
5119  013462                                ERROR   26              ;ERROR IR IS NOT CLEAR
 (5)  013500  104455                        TRAP    C$ERDF
 (6)  013502  000032                        .WORD   26
 (6)  013504  005474                        .WORD   EM26
 (6)  013506  007566                        .WORD   ERR26
5120  013510                        2$:     ESCAPE  SEG
 (3)  013510  104410                        TRAP    C$ESCAPE
 (3)  013512  000002                        .WORD   10000$-.
5121  013514                                ENDSEG
 (3)  013514                        10000$:
 (3)  013514  104405                        TRAP    C$ESEG
5122  013516  012702  177777                MOV     #-1,R2          ;PUT 'EXPECTED'' IN $GDDAT
5123  013522                                BGNSEG
 (3)  013522  104404                        TRAP    C$BSEG
5124  013524  010261  000006        3$:     MOV     R2,6(R1)        ;WRITE ALL ONES TO THE IR
5125  013530  016104  000006                MOV     6(R1),R4        ;READ THE IR
5126  013534  020204                        CMP     R2,R4           ;IS IT ALL ONES?
```

```
5127  013536  001413                    BEQ     4$              ;BR IF YES
5128  013540                            ERROR   26              ;ERROR IR IS NOT - ALL ONES
 (5)  013556  104455                    'RAP    C$ERDF
 (6)  013560  000032                    .WORD   26
 (6)  013562  005474                    .WORD   EM26
 (6)  013564  007566                    .WORD   ERR26
5129  013566                    4$:     ESCAPE  SEG
 (3)  013566  104410                    TRAP    C$ESCAPE
 (3)  013570  000002                    .WORD   1000'$-.
5130  013572                            ENDSEG
 (3)  013572                    10001$:
 (3)  013572  104405                    TRAP    C$ESEG
5131  013574                    ENDTST
 (3)  013574                    L10065:
 (3)  013574  104401                    TRAP    'SE'S'
5132
5133
```

```
 5135
 5136  013576                                  BADHEAD
  (2)                                          ;******************** TEST 13 ********************
 5137                                          ;*MICRO PROCESSOR TEST
 5138                                          ;*LOAD KMP06 WITH A MICRO-PROCESSOR INSTRUCTION, CLOCK IT
 5139                                          ;*VERIFY INSTRUCTION EXECUTED PROPERLY
 5140                                          ;*INSTRUCTION SHOULD MOVE IBUS 4 TO IBUS*5, IBUS*4 IS ALL 1'S
 5141                                          ;*AND IBUS*5 IS ALL 0'S. RESULT SHOULD BE ALL 1'S IN SEL4
 5142  013576                                  BADHEAD
  (2)                                          ;******************** TEST 13 ********************
 5143
 5144  013576                            BGNTST
  (3)  013576                            T13::
 5145  013576                                  MYINT
  (1)  013576  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 5146  013602                                  MSTCLR
  (1)  013602  004537  003156                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 5147  013606  012761  000377  000004        MOV     #377,4(R1)      ;PORT4 HI BYTE-1'S
 5148  013614  012711  001000                MOV     #BIT9,(R1)              ;SET ROMI
 5149  013620  012761  121105  000006        MOV     #121105,6(R1)   ;INSTR TO PORT 6.
 5150  013626  052711  001400                BIS     #BIT8!BIT9,(R1) ;CLK INSTR.
 5151  013632  000240                        NOP
 5152  013634  012702  177777                MOV     #-1,R2          ;EXPECT ALL ONES.
 5153  013640  116104  000004                MOVB    4(R1),R4        ;READ FOUND.
 5154  013644  020204                        CMP     R2,R4   ;DATA CORRECT?
 5155  013646  001413                        BEQ     1$
 5156  013650                                 ERROR   28
  (5)  013666  104455                        TRAP    C$ERDF
  (6)  013670  000034                        .WORD   28
  (6)  013672  005554                        .WORD   EM28
  (6)  013674  007726                        .WORD   ERR28
 5157
 5158  013676                            1$:     ESCAPE  TST
  (3)  013676  104410                        TRAP    C$ESCAPE
  (3)  013700  000002                        .WORD   L10066-.
 5159
 5160  013702                            ENDTST
  (3)  013702                            L10066:
  (3)  013702  104401                        TRAP    C$ETST
 5161
 5162  013704                                  BADHEAD
  (2)                                          ;******************** TEST 14 ********************
 5163                                          ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
 5164                                          ;*FLOAT A 1 THOUGH IBUS* REGISTER 0
 5165                                          ;*FLOAT A 0 THROUGH IBUS* REGISTER 0
 5166  013704                                  BADHEAD
  (2)                                          ;******************** TEST 14 ********************
 5167
 5168  013704                            BGNTST
  (3)  013704                            T14::
 5169  013704                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  013704  004537  003156                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 5170  013710  012737  000000  002624        MOV     #0,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
 5171  013716  012705  000001                MOV     #1,R5           ;START WITH BIT 0
 5172
 5173  013722                                  MYINT
```

```
  (1)  013722  013701  002716            MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5174   013726                             BGNSEG
  (3)  013726  104404                     TRAP    C$BSEG
5175   013730                     64$:
5176   013730  010561  000004             MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5177   013734                             ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  013734  004537  003244             JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5178   013740  121100                     121100                  ;MOV DATA TO IBUS* REGISTER 0
5179   013742                             ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  013742  004537  003244             JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5180   013746  121005                     121005                  ;READ FROM IBU* REGITER 0
5181   013750  116104  000005             MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
5182   013754  120504                     CMPB    R5,R4           ;DATA CORRECT?
5183   013756  001414                     BEQ     65$             ;BR IF YES
5184   013760                             BERROR  27              ;ERROR
  (5)  014000  104455                     TRAP    C$ERDF
  (6)  014002  000033                     .WORD   27
  (6)  014004  005525                     .WORD   EM27
  (6)  014006  007644                     .WORD   ERR27
5185   014010                     65$:    ESCAPE  SEG
  (3)  014010  104410                     TRAP    C$ESCAPE
  (3)  014012  000010                     .WORD   10000$-.
5186   014014  000241                     CLC                     ;CLEAR CARRY
5187   014016  106105                     ROLB    R5              ;SHIFT BIT IN R5
5188   014020  001343                     BNE     64$             ;IF R2=0 THEN DONE
5189   014022                             ENDSEG
  (3)  014022                     10000$:
  (3)  014022  104405                     TRAP    C$ESEG
5190   014024  012705  000001             MOV     #1,R5           ;START WITH BIT 0
5191                             ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
5192   014030                             BGNSEG
  (3)  014030  104404                     TRAP    C$BSEG
5193   014032                     6/$:
5194   014032  005105                     COM     R5
5195   014034  010561  000004             MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5196   014040                             ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  014040  004537  003244             JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5197   014044  121100                     121100                  ;MOV DATA TO IBUS* REGISTER 0
5198   014046                             ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  014046  004537  003244             JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5199   014052  121005                     121005                  ;READ FROM IBUS* REGISTER 0
5200   014054  116104  000005             MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
5201   014060  120504                     CMPB    R5,R4           ;DATA CORRECT?
5202   014062  001414                     BEQ     68$             ;BR IF YES
5203   014064                             BERROR  27              ;ERROR
  (5)  014104  104455                     TRAP    C$ERDF
  (6)  014106  000033                     .WORD   27
  (6)  014110  005525                     .WORD   EM27
  (6)  014112  007644                     .WORD   ERR27
5204   014114                     68$:    ESCAPE  SEG
  (3)  014114  104410                     TRAP    C$ESCAPE
  (3)  014116  000012                     .WORD   10001$-.
5205   014120  005105                     COM     R5              ;CHANGE TO FLOATING 1
5206   014122  000241                     CLC                     ;CLEAR CARRY
5207   014124  106105                     ROLB    R5              ;SHIFT BIT IN R5
5208   014126  001341                     BNE     67$             ;IF R2=0 THEN DONE
```

```
      014130                              ENDSEG
(3)   014130                     10001$:
(3)   014130  104405                      TRAP    C$ESEG
      014132                     ENDTST
(3)   014132                     L10067:
(3)   014132  '04401                      TRAP    C$ETST

      014134                              BADHEAD
(2)                                       ;******************** TEST 15 ********************
                                          ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
                                          ;*FLOAT A 1 THOUGH IBUS* REGISTER 2
                                          ;*FLOAT A 0 THROUGH IBUS* REGISTER 2
      014134                              BADHEAD
(2)                                       ;******************** TEST 15 ********************

      014134                     BGNTST
(3)   014134                     T15::
      014134                              MSTCLR                   ;MASTER CLEAR M8200,4,7
(1)   014134  004537  003156              JSR     R5,.MSTCLR               ;CLEAR M8200,4,7
      014140  012737  000002  002624      MOV     #2,MR0           ;SAVE REGISTER ADDRESS FOR TYPEOUT
      014146  012705  000001              MOV     #1,R5            ;START WITH BIT 0
      014152                              MYINT
(1)   014152  013701  002716              MOV     KMCSR,R1         ;GET DEVICE ADDRESS.
      014156                              BGNSEG
(3)   014156  104404                      TRAP    C$BSEG
      014160                     64$:
      014160  010561  000004              MOV     R5,4(R1)         ;PUT PATTERN INTO PORT4
      014164                              ROMCLK                   ;NEXT WORD IS INSTRUCTION.  BBN
(1)   014164  004537  003244              JSR     R5,.ROMCLK               ;CLOCK INSTRUCTION
      014170  121102                      121100!2                 ;MOV DATA TO IBUS* REGISTER 0
      014172                              ROMCLK                   ;NEXT WORD IS INSTRUCTION.  BBN
(1)   014172  004537  003244              JSR     R5,.ROMCLK               ;CLOCK INSTRUCTION
      014176  121045                      121005!<2*20>            ;READ FROM IBUS* REGISTER 2
      014200  116104  000005              MOVB    5(R1),R4         ;PUT ''FOUND'' INTO R4
      014204  120504                      CMPB    R5,R4            ;DATA CORRECT?
      014206  001414                      BEQ     65$              ;BR IF YES
      014210                              BERROR  27               ;ERROR
(5)   014230  104455                      TRAP    C$ERDF
(6)   014232  000033                      .WORD   27
(6)   014234  005525                      .WORD   EM27
(6)   014236  007644                      .WORD   ERR27
      014240                     65$:     ESCAPE  SEG
(3)   014240  104410                      TRAP    C$ESCAPE
(3)   014242  000010                      .WORD   10000$-.
      014244  000241                      CLC                      ;CLEAR CARRY
      014246  106105                      ROLB    R5               ;SHIFT BIT IN R2
      014250  001343                      BNE     64$              ;IF R2-0 THEN DONE
      014252                              ENDSEG
(3)   014252                     10000$:
(3)   014252  104405                      TRAP    C$ESEG
      014254  012705  000001              MOV     #1,R5            ;START WITH BIT 0
                                 ;69$:    COM     R5               ;CHANGE TO FLOATING ZERO
      014260                              BGNSEG
(3)   014260  104404                      TRAP    C$BSEG
      014262                     67$:
      014262  005105                      COM     R5
```

```
5244  014264  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5245  014270                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (*)  014270  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5246  014274  121102                          121100!2                ;MOV DATA TO IBUS* REGISTER 2
5247  014276                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  014276  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5248  014302  121005                          121005!<2*20>           ;READ FROM IBUS* REGISTER 2
5249  014304  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
5250  014310  120504                          CMPB    R5,R4           ;DATA CORRECT?
5251  014312  001414                          BEQ     68$             ;BR IF YES
5252  014314                                  BERROR  27              ;ERROR
 (5)  014334  104455                          TRAP    C$ERDF
 (6)  014336  000033                          .WORD   27
 (6)  014340  005525                          .WORD   EM27
 (6)  014342  007644                          .WORD   ERR27
5253  014344                          68$:    ESCAPE  SEG
 (3)  014344  104410                          TRAP    C$ESCAPE
 (3)  014346  000012                          .WORD   10001$-.
5254  014350  005105                          COM     R5              ;CHANGE TO FLOATING 1
5255  014352  000241                          CLC                     ;CLEAR CARRY
5256  014354  106105                          ROLB    R5              ;SHIFT BIT IN R2
5257  014356  001341                          BNE     67$             ;IF R2=0 THEN DONE
5258  014360                                  ENDSEG
 (3)  014360                          10001$:
 (3)  014360  104405                          TRAP    C$ESEG
5259  014362                          ENDTST
 (3)  014362                          L10070:
 (3)  014362  104401                          TRAP    C$ETST
5260
5261  014364                                  BADHEAD
 (2)                                      ;****************** TEST 16 ******************
5262                                       ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
5263                                       ;*FLOAT A 1 THOUGH IBUS* REGISTER 4
5264                                       ;*FLOAT A 0 THROUGH IBUS* REGISTER 4
5265  014364                                  BADHEAD
 (2)                                      ;****************** TEST 16 ******************
5266
5267  014364                          BGNTST
 (3)  014364                          T16::
5268  014364                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  014364  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
5269  014370  012737  000004  002624          MOV     #4,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
5270  014376  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
5271  014402                                  MYINT
 (1)  014402  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5272  014406                                  BGNSEG
 (3)  014406  104404                          TRAP    C$BSEG
5273  014410                          64$:
5274  014410  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5275  014414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  014414  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5276  014420  121104                          121100!4                ;MOV DATA TO IBUS* REGISTER 4
5277  014422                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  014422  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5278  014426  121105                          121005!<4*20>           ;READ FROM IBUS* REGISTER 4
5279  014430  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
```

```
 5280  014434  120504                        CMPB    R5,R4           ;DATA CORRECT?
 5281  014436  001414                        BEQ     65$             ;BR IF YES
 5282  014440                                BERROR  27              ;ERROR
  (5)  014460  104455                        TRAP    C$ERDF
  (6)  014462  000033                        .WORD   27
  (6)  014464  005525                        .WORD   EM27
  (6)  014466  007644                        .WORD   ERR27
 5283  014470                          65$:   ESCAPE  SEG
  (3)  014470  104410                        TRAP    C$ESCAPE
  (3)  014472  000010                        .WORD   10000$-.
 5284  014474  000241                        CLC                     ;CLEAR CARRY
 5285  014476  106105                        ROLB    R5              ;SHIFT BIT IN R2
 5286  014500  001343                        BNE     64$             ;IF R2-0 THEN DONE
 5287  014502                                ENDSEG
  (3)  014502                          10000$:
  (3)  014502  104405                        TRAP    C$ESEG
 5288  014504  012705  000001                MOV     #1,R5           ;START WITH BIT 0
 5289                                ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
 5290  014510                                BGNSEG
  (3)  014510  104404                        TRAP    C$BSEG
 5291  014512                          67$:
 5292  014512  005105                        COM     R5
 5293  014514  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 5294  014520                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  014520  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5295  014524  121104                        121100!4                ;MOV DATA TO IBUS* REGISTER 4
 5296  014526                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  014526  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5297  014532  121105                        121005!<4*20>           ;READ FROM IBUS* REGISTER 4
 5298  014534  116104  000005                MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 5299  014540  120504                        CMPB    R5,R4           ;DATA CORRECT?
 5300  014542  001414                        BEQ     68$             ;BR IF YES
 5301  014544                                BERROR  27              ;ERROR
  (5)  014564  104455                        TRAP    C$ERDF
  (6)  014566  000033                        .WORD   27
  (6)  014570  005525                        .WORD   EM27
  (6)  014572  007644                        .WORD   ERR27
 5302  014574                          68$:   ESCAPE  SEG
  (3)  014574  104410                        TRAP    C$ESCAPE
  (3)  014576  000012                        .WORD   10001$-.
 5303  014600  005105                        COM     R5              ;CHANGE TO FLOATING 1
 5304  014602  000241                        CLC                     ;CLEAR CARRY
 5305  014604  106105                        ROLB    R5              ;SHIFT BIT IN R2
 5306  014606  001341                        BNE     67$             ;IF R2=0 THEN DONE
 5307  014610                                ENDSEG
  (3)  014610                          10001$:
  (3)  014610  104405                        TRAP    C$ESEG
 5308  014612                          ENDTST
  (3)  014612                          L10071:
  (3)  014612  104401                        TRAP    C$ETST
 5309
 5310  014614                                BADHEAD
  (2)                                   ;********************* TEST 17 *********************
 5311                                   ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
 5312                                   ;*FLOAT A 1 THOUGH IBUS* REGISTER 5
 5313                                   ;*FLOAT A 0 THROUGH IBUS* REGISTER 5
```

```
                                        BADHEAD
                                        ;****************** TEST 17 ******************

                                BGNTST
                                T17::
                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
(1)  014614  004537  003156             JSR     R5,,MSTCLR              ;CLEAR M8200,4,7
5318 014620  012737  000005  002624     MOV     #5,MRO          ;SAVE REGISTER ADDRESS FOR TYPEOUT
5319 014626  012705  00000'             MOV     #1,R5           ;START WITH BIT 0
5320 014632                             MYINT
(1)  014632  013701  002716             MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5321 014636                             BGNSEG
(3)  014636  104404                     TRAP    C$BSEG
5322 014640                     64$:
5323 014640  010561  000004             MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5324 014644                             ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
(1)  014644  004537  003244             JSR     R5,,ROMCLK              ;CLOCK INSTRUCTION
5325 014650  121105                     121100!5                ;MOV DATA TO IBUS* REGISTER 5
5326 014652                             ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
(1)  014652  004537  003244             JSR     R5,,ROMCLK              ;CLOCK INSTRUCTION
5327 014656  121125                     121005!<5*20>           ;READ FROM IBUS* REGISTER 5
5328 014660  116104  000005             MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
5329 014664  120504                     CMPB    R5,R4           ;DATA CORRECT?
5330 014666  001414                     BEQ     65$             ;BR IF YES
5331 014670                             BERROR  27              ;ERROR
(5)  014710  104455                     TRAP    C$ERDF
(6)  014712  000033                     .WORD   27
(6)  014714  005525                     .WORD   EM27
(6)  014716  007644                     .WORD   ERR27
5332 014720                     65$:    ESCAPE  SEG
(3)  014720  104410                     TRAP    C$ESCAPE
(3)  014722  000010                     .WORD   10000$-.
5333 014724  000241                     CLC                     ;CLEAR CARRY
5334 014726  106105                     ROLB    R5              ;SHIFT BIT IN R5
5335 014730  001343                     BNE     64$             ;IF R5=0 THEN DONE
5336 014732                             ENDSEG
(3)  014732                     10000$:
(3)  014732  104405                     TRAP    C$ESEG
5337 014734  012705  000001             MOV     #1,R5           ;START WITH BIT 0
5338                            ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
5339 014740                             BGNSEG
(3)  014740  104404                     TRAP    C$BSEG
5340 014742                     67$:
5341 014742  005105                     COM     R5
5342 014744  010561  000004             MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5343 014750                             ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
(1)  014750  004537  003244             JSR     R5,,ROMCLK              ;CLOCK INSTRUCTION
5344 014754  121105                     121100!5                ;MOV DATA TO IBUS* REGISTER 5
5345 014756                             ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
(1)  014756  004537  003244             JSR     R5,,ROMCLK              ;CLOCK INSTRUCTION
5346 014762  121125                     121005.<5*20>           ;READ FROM IBUS* REGISTER 5
5347 014764  116104  000005             MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
5348 014770  120504                     CMPB    R5,R4           ;DATA CORRECT?
5349 014772  001414                     BEQ     68$             ;BR IF YES
5350 014774                             BERROR  27              ;ERROR
(5)  015014  104455                     TRAP    C$ERDF
```

```
 (6)  015016  000033                        .WORD   27
 (6)  015020  005525                        .WORD   EM27
 (6)  015022  007644                        .WORD   ERR27
5351  015024                        68$:    ESCAPE  SEG
 (3)  015024  104410                        TRAP    C$ESCAPE
 (3)  015026  000012                        .WORD   10001$-.
5352  015030  005105                        COM     R5              ;CHANGE TO FLOATING 1
5353  015032  000241                        CLC                     ;CLEAR CARRY
5354  015034  106105                        ROLB    R5              ;SHIFT BIT IN R5
5355  015036  001341                        BNE     67$             ;IF R5-0 THEN DONE
5356  015040                                ENDSEG
 (3)  015040                        10001$:
 (3)  015040  104405                        TRAP    C$ESEG
5357  015042                        ENDTST
 (3)  015042                        L10072:
 (3)  015042  104401                        TRAP    C$ETST
5358
5359  015044                                BADHEAD
 (2)                                        ;****************** TEST 18 ******************
5360                                        ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
5361                                        ;*FLOAT A 1 THOUGH IBUS* REGISTER 10
5362                                        ;*FLOAT A 0 THROUGH IBUS* REGISTER 10
5363  015044                                BADHEAD
 (2)                                        ;****************** TEST 18 ******************
5364
5365  015044                        BGNTST
 (3)  015044                        T18::
5366  015044                                MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  015044  004537  003156                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
5367  015050  012737  000010  002624        MOV     #10,MR0         ;SAVE REGISTER ADDRESS FOR TYPEOUT
5368  015056  012705  000001                MOV     #1,R5           ;START WITH BIT 0
5369  015062                                MYINT
 (1)  015062  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5370  015066                                BGNSEG
 (3)  015066  104404                        TRAP    C$BSEG
5371  015070                        64$:
5372  015070  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5373  015074  042761  000141  000004        BIC     #141,4(R1)      ;CLEAR UNWANTED BITS
5374  015102                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  015102  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5375  015106  121110                        121100!^O               ;MOV DATA TO IBUS* REGISTER 10
5376  015110                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  015110  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5377  015114  121205                        121005!<10*20>          ;READ FROM IBUS* REGISTER 10
5378  015116  010502                        MOV     R5,R2
5379  015120  042705  000141                BIC     #141,R5         ;CLEAR UNWANTED BITS
5380  015124  116104  000005                MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
5381  015130  042704  000140                BIC     #140,R4         ;CLEAR UNWANTED BITS
5382  015134  120504                        CMPB    R5,R4           ;DATA CORRECT?
5383  015136  001414                        BEQ     65$             ;BR IF YES
5384  015140                                BERROR  27              ;ERROR
 (5)  015160  104455                        TRAP    C$ERDF
 (6)  015162  000033                        .WORD   27
 (6)  015164  005525                        .WORD   EM27
 (6)  015166  007644                        .WORD   ERR27
5385  015170                        65$:    ESCAPE  SEG
```

```
   (3)  015170  10441C                          TRAP    C$ESCAPE
   (3)  015172  000012                          .WORD   10000$-.
 5386   015174  010205                          MOV     R2,R5
 5387   015176  000241                          CLC                     ;CLEAR CARRY
 5388   015200  106105                          ROLB    R5              ;SHIFT BIT IN R5
 5389   015202  001332                          BNE     64$             ;IF R5=0 THEN DONE
 5390   015204                                  ENDSEG
   (3)  015204                          10000$:
   (3)  015204  104405                          TRAP    C$ESEG
 5391   015206  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
 5392                               ;69$:       COM     R5              ;CHANGE TO FLOATING ZERO
 5393   015212                                  BGNSEG
   (3)  015212  104404                          TRAP    C$BSEG
 5394   015214                          67$:
 5395   015214  005105                          COM     R5
 5396   015216  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 5397   015222  042761  000141  000004          BIC     #141,4(R1)      ;CLEAR UNWANTED BITS
 5398   015230                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION.  BBN
   (1)  015230  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5399   015234  121110                          121100!10               ;MOV DATA TO IBUS* REGISTER 10
 5400   015236                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION.  BBN
   (1)  015236  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5401   015242  121205                          121005!<10*20>          ;READ FROM IBUS* REGISTER 10
 5402   015244  010502                          MOV     R5,R2
 5403   015246  042705  000141                  BIC     #141,R5         ;CLEAR UNWANTED BITS
 5404   015252  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 5405   015256  042704  000140                  BIC     #140,R4         ;CLEAR UNWANTED BITS
 5406   015262  120504                          CMPB    R5,R4           ;DATA CORRECT?
 5407   015264  001414                          BEQ     68$             ;BR IF YES
 5408   015266                                  BERROR  27              ;ERROR
   (5)  015306  104455                          TRAP    C$ERDF
   (6)  015310  000033                          .WORD   27
   (6)  015312  005525                          .WORD   EM27
   (6)  015314  007644                          .WORD   ERR27
 5409   015316                          68$:    ESCAPE  SEG
   (3)  015316  104410                          TRAP    C$ESCAPE
   (3)  015320  000014                          .WORD   10001$-.
 5410   015322  010205                          MOV     R2,R5
 5411   015324  005105                          COM     R5              ;CHANGE TO FLOATING 1
 5412   015326  000241                          CLC                     ;CLEAR CARRY
 5413   015330  106105                          ROLB    R5              ;SHIFT BIT IN R5
 5414   015332  001330                          BNE     67$             ;IF R5-0 THEN DONE
 5415   015334                                  ENDSEG
   (3)  015334                          10001$:
   (3)  015334  104405                          TRAP    C$ESEG
 5416   015336                          ENDTST
   (3)  015336                          L10073:
   (3)  015336  104401                          TRAP    C$ETST
 5417
 5418   015340                                  BADHEAD
   (2)                                           ;****************** TEST 19 ******************
 5419                                            ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
 5420                                            ;*FLOAT A 1 THOUGH IBUS* REGISTER 11
 5421                                            ;*FLOAT A 0 THROUGH IBUS* REGISTER 11
 5422   015340                                  BADHEAD
   (2)                                           ;****************** TEST 19 ******************
```

```
5423
5424  015340                                  BGNTST
 (3)  015340                                  T19::
5425  015340                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  015340  004537  003156                         JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
5426  015344  012737  000011  002624                MOV     #11,MR0         ;SAVE REGISTER ADDRESS FOR TYPEOUT
5427  015352  012705  000001                       MOV     #1,R5           ;START WITH BIT 0
5428  015356                                       MYINT
 (1)  015356  013701  002716                       MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5429  015362                                       BGNSEG
 (3)  015362  104404                              TRAP    C$BSEG
5430  015364                                  64$:
5431  015364  010561  000004                       MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5432  015370  042761  000262  000004               BIC     #262,4(R1)      ;CLEAR UNWANTED BITS
5433  015376                                       ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  015376  004537  003244                         JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5434  015402  121111                               121100.11               ;MOV DATA TO IBUS* REGISTER 11
5435  015404                                       ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  015404  004537  003244                         JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5436  015410  121225                               121005!<11*20>          ;READ FROM IBUS* REGISTER 11
5437  015412  010502                               MOV     R5,R2
5438  015414  042705  000262                       BIC     #262,R5         ;CLEAR UNWANTED BITS
5439  015420  116104  000005                       MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
5440  015424  042704  000020                       BIC     #20,R4
5441  015430  120504                               CMPB    R5,R4           ;DATA CORRECT?
5442  015432  001414                               BEQ     65$             ;BR IF YES
5443  015434                                       BERROR  27              ;ERROR
 (5)  015454  104455                              TRAP    C$ERDF
 (6)  015456  000033                              .WORD   27
 (6)  015460  005525                              .WORD   EM27
 (6)  015462  007644                              .WORD   ERR27
```

```
 5445  015464                                 65$:    ESCAPE  SEG
  (3)  015464  104410                                TRAP    C$ESCAPE
  (3)  015466  000012                                .WORD   10000$-.
 5446  015470  010205                                MOV     R2,R5
 5447  015472  000241                                CLC                     ;CLEAR CARRY
 5448  015474  106105                                ROLB    R5              ;SHIFT BIT IN R5
 5449  015476  001332                                BNE     64$             ;IF R5=0 THEN DONE
 5450  015500                                         ENDSEG
  (3)  015500                                 10000$:
  (3)  015500  104405                                TRAP    C$ESEG
 5451  015502  012705  000001                        MOV     #1,R5           ;START WITH BIT 0
 5452                                      ;69$:    COM     R5              ;CHANGE TO FLOATING ZERO
 5453  015506                                         BGNSEG
  (3)  015506  104404                                TRAP    C$BSEG
 5454  015510                                 67$:
 5455  015510  005105                                COM     R5
 5456  015512  010561  000004                        MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 5457  015516  042761  000262  000004                BIC     #262,4(R1)      ;CLEAR UNWANTED BITS
 5458  015524                                         ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  015524  004537  003244                        JSR     R5,,ROMCLK              ;CLOCK INSTRUCTION
 5459  015530  121111                                121100.11               ;MOV DATA TO IBUS* REGISTER 11
 5460  015532                                         ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  015532  004537  003244                        JSR     R5,,ROMCLK              ;CLOCK INSTRUCTION
 5461  015536  121225                                121005:<11*20>          ;READ FROM IBUS* REGISTER 11
```

```
5463  015540  010502                          MOV     R5,R2
5464  015542  042705  000262                  BIC     #262,R5         ;CLEAR UNWANTED BITS
5465  015546  052705  000020                  BIS     #20,R5          ;ADD THESE BITS
5466  015552  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
5467  015556  120504                          CMPB    R5,R4           ;DATA CORRECT?
5468  015560  001414                          BEQ     68$             ;BR IF YES
5469  015562                                  BERROR  27              ;ERROR
 (5)  015602  104455                          TRAP    C$ERDF
 (6)  015604  000033                          .WORD   27
 (6)  015606  005525                          .WORD   EM27
 (6)  015610  007644                          .WORD   ERR27
5470  015612                          68$:    ESCAPE  SEG
 (3)  015612  104410                          TRAP    C$ESCAPE
 (3)  015614  000014                          .WORD   10001$-.
5471  015616  010205                          MOV     R2,R5
5472  015620  005105                          COM     R5              ;CHANGE TO FLOATING 1
5473  015622  000241                          CLC                     ;CLEAR CARRY
5474  015624  106105                          ROLB    R5              ;SHIFT BIT IN R5
5475  015626  001330                          BNE     67$             ;IF R5-0 THEN DONE
5476  015630                                  ENDSEG
 (3)  015630                          10001$:
 (3)  015630  104405                          TRAP    C$ESEG
5477  015632                          ENDTST
 (3)  015632                          L10074:
 (3)  015632  104401                          TRAP    C$ETST
5478
5479  015634                                  BADHEAD
 (2)                                          ;****************** TEST 20 ******************
5480                                          ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
5481                                          ;*FLOAT A 1 THOUGH IBUS REGISTER 0
5482                                          ;*FLOAT A 0 THROUGH IBUS REGISTER 0
5483  015634                                  BADHEAD
 (2)                                          ;****************** TEST 20 ******************
5484
5485  015634                          BGNTST
 (3)  015634                          T20::
5486  015634                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  015634  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
5487  015640  012737  000000  002624          MOV     #0,MRC          ;SAVE REGISTER ADDRESS FOR TYPEOUT
5488  015646  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
5489  015652                                  MYINT
 (1)  015652  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5490  015656                                  BGNSEG
 (3)  015656  104404                          TRAP    C$BSEG
5491  015660                          64$:
5492  015660  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5493  015664                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  015664  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5494  015670  122100                          122100                  ;MOV DATA TO IBUS* REGISTER 0
5495  015672                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  015672  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5496  015676  021005                          21005                   ;READ FROM IBUS* REGISTER 0
5497  015700  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
5498  015704  120504                          CMPB    R5,R4           ;DATA CORRECT?
5499  015706  001414                          BEQ     65$             ;BR IF YES
5500  015710                                  BERROR  29              ;ERROR
```

```
 (5)  015730  104455                        TRAP    C$ERDF
 (6)  015732  000035                        .WORD   29
 (6)  015734  005605                        .WORD   EM29
 (6)  015736  010004                        .WORD   ERR29
5501  015740                        65$:    ESCAPE  SEG
 (3)  015740  104410                        TRAP    C$ESCAPE
 (3)  015742  000010                        .WORD   10000$-.
5502  015744  000241                        CLC                     ;CLEAR CARRY
5503  015746  106105                        ROLB    R5              ;SHIFT BIT IN R5
5504  015750  001343                        BNE     64$             ;IF R5=0 THEN DONE
5505  015752                                ENDSEG
 (3)  015752                        10000$:
 (3)  015752  104405                        TRAP    C$ESEG
5506  015754  012705  000001              MOV     #1,R5           ;START WITH BIT 0
5507                              ;69$:     COM     R5              ;CHANGE TO FLOATING ZERO
5508  015760                              BGNSEG
 (3)  015760  104404                        TRAP    C$BSEG
5509  015762                        67$:
5510  015762  005105                        COM     R5
5511  015764  010561  000004              MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5512  015770                              ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  015770  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5513  015774  122100                      122100                  ;MOV DATA TO IBUS* REGISTER 0
5514  015776                              ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  015776  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5515  016002  021005                      21005                   ;READ FROM IBUS* REGISTER 0
5516  016004  116104  000005              MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
5517  016010  120504                      CMPB    R5,R4           ;DATA CORRECT?
5518  016012  001414                      BEQ     68$             ;BR IF YES
5519  016014                              BERROR  29              ;ERROR
 (5)  016034  104455                        TRAP    C$ERDF
 (6)  016036  000035                        .WORD   29
 (6)  016040  005605                        .WORD   EM29
 (6)  016042  010004                        .WORD   ERR29
5520  016044                        68$:    ESCAPE  SEG
 (3)  016044  104410                        TRAP    C$ESCAPE
 (3)  016046  000012                        .WORD   10001$-.
5521  016050  005105                        COM     R5              ;CHANGE TO FLOATING 1
5522  016052  000241                        CLC                     ;CLEAR CARRY
5523  016054  106105                        ROLB    R5              ;SHIFT BIT IN R5
5524  016056  001341                        BNE     67$             ;IF R5=0 THEN DONE
5525  016060                                ENDSEG
 (3)  016060                        10001$:
 (3)  016060  104405                        TRAP    C$ESEG
5526  016062                        ENDTST
 (3)  016062                        L10075:
 (3)  016062  104401                        TRAP    C$ETST
5527
5528  016064                              BADHEAD
 (2)                                    ;****************** TEST 21 ******************
5529                                    ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
5530                                    ;*FLOAT A 1 THOUGH IBUS REGISTER 1
5531                                    ;*FLOAT A 0 THROUGH IBUS REGISTER 1
5532  016064                              BADHEAD
 (2)                                    ;****************** TEST 21 ******************
5533
```

```
5534  016064                                  BGNTST
 (3)  016064                                  T21::
5535  016064                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  016064  004537  003156                        JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
5536  016070  012737  000001  002624                MOV     #1,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
5537  016076  012705  000001                      MOV     #1,R5           ;START WITH BIT 0
5538  016102                                      MYINT
 (1)  016102  013701  002716                      MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5539  016106                                      BGNSEG
 (3)  016106  104404                              TRAP    C$BSEG
5540  016110                                64$:
5541  016110  010561  000004                      MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5542  016114                                      ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  016114  004537  003244                      JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5543  016120  122101                            122100!1                ;MOV DATA TO IBUS* REGISTER 1
5544  016122                                      ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  016122  004537  003244                      JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5545  016126  021025                            21005!<1*20>            ;READ FROM IBUS* REGISTER 1
5546  016130  116104  000005                      MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
5547  016134  120504                            CMPB    R5,R4           ;DATA CORRECT?
5548  016136  001414                            BEQ     65$             ;BR IF YES
5549  016140                                      BERROR  29              ;ERROR
 (5)  016160  104455                              TRAP    C$ERDF
 (6)  016162  000035                              .WORD   29
 (6)  016164  005605                              .WORD   EM29
 (6)  016166  010004                              .WORD   ERR29
5550  016170                                65$:  ESCAPE  SEG
 (3)  016170  104410                              TRAP    C$ESCAPE
 (3)  016172  000010                              .WORD   10000$-.
5551  016174  000241                            CLC                     ;CLEAR CARRY
5552  016176  106105                            ROLB    R5              ;SHIFT BIT IN R5
5553  016200  001343                            BNE     64$             ;IF R5=0 THEN DONE
5554  016202                                      ENDSEG
 (3)  016202                                10000$:
 (3)  016202  104405                              TRAP    C$ESEG
5555  016204  012705  000001                      MOV     #1,R5           ;START WITH BIT 0
5556                                        ;69$: COM     R5              ;CHANGE TO FLOATING ZERO
5557  016210                                      BGNSEG
 (3)  016210  104404                              TRAP    C$BSEG
5558  016212                                67$:
5559  016212  005105                            COM     R5
5560  016214  010561  000004                      MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5561  016220                                      ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  016220  004537  003244                      JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5562  016224  122101                            122100!1                ;MOV DATA TO IBUS* REGISTER 1
5563  016226                                      ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  016226  004537  003244                      JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5564  016232  021025                            21005!<1*20>            ;READ FROM IBUS* REGISTER 1
5565  016234  116104  000005                      MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
5566  016240  120504                            CMPB    R5,R4           ;DATA CORRECT?
5567  016242  001414                            BEQ     68$             ;BR IF YES
5568  016244                                      BERROR  29              ;ERROR
 (5)  016264  104455                              TRAP    C$ERDF
 (6)  016266  000035                              .WORD   29
 (6)  016270  005605                              .WORD   EM29
 (6)  016272  010004                              .WORD   ERR29
```

```
 5569  016274                          68$:    ESCAPE  SEG
  (3)  016274  104410                         TRAP    C$ESCAPE
  (3)  016276  000012                         .WORD   10001$-.
 5570  016300  005105                         COM     R5              ;CHANGE TO FLOATING 1
 5571  016302  000241                         CLC                     ;CLEAR CARRY
 5572  016304  106105                         ROLB    R5              ;SHIFT BIT IN R5
 5573  016306  001341                         BNE     67$             ;IF R5-0 THEN DONE
 5574  016310                                 ENDSEG
  (3)  016310                          10001$:
  (3)  016310  104405                         TRAP    C$ESEG
 5575  016312                          ENDTST
  (3)  016312                          L10076:
  (3)  016312  104401                         TRAP    C$ETST
 5576
 5577  016314                                 BADHEAD
  (2)                                        ;****************** TEST 22 ******************
 5578                                        ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
 5579                                        ;*FLOAT A 1 THOUGH IBUS REGISTER 2
 5580                                        ;*FLOAT A 0 THROUGH IBUS REGISTER 2
 5581  016314                                 BADHEAD
  (2)                                        ;****************** TEST 22 ******************
 5582
 5583  016314                          BGNTST
  (3)  016314                          T22::
 5584  016314                                 MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  016314  004537  003156                 JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 5585  016320  012737  000002  002624         MOV     #2,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
 5586  016326  012705  000001                 MOV     #1,R5           ;START WITH BIT 0
 5587  016332                                 MYINT
  (1)  016332  013701  002716                 MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 5588  016336                                 BGNSEG
  (3)  016336  104404                         TRAP    C$BSEG
 5589  016340                          64$:
 5590  016340  010561  000004                 MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 5591  016344                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  016344  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 5592  016350  122102                         122100!2                ;MOV DATA TO IBUS* REGISTER 2
 5593  016352                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  016352  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 5594  016356  021045                         21005!<2*20>            ;READ FROM IBUS* REGISTER 2
 5595  016360  116104  000005                 MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
 5596  016364  120504                         CMPB    R5,R4           ;DATA CORRECT?
 5597  016366  001404                         BEQ     65$             ;BR IF YES
 5598  016370                                 BERROR  29              ;ERROR
  (5)  016410  104455                         TRAP    C$ERDF
  (6)  016412  000035                         .WORD   29
  (6)  016414  005605                         .WORD   EM29
  (6)  016416  010004                         .WORD   ERR29
 5599  016420                          65$:    ESCAPE  SEG
  (3)  016420  104410                         TRAP    C$ESCAPE
  (3)  016422  000010                         .WORD   10000$-.
 5600  016424  000241                         CLC                     ;CLEAR CARRY
 5601  016426  106105                         ROLB    R5              ;SHIFT BIT IN R5
 5602  016430  001343                         BNE     64$             ;IF R5=0 THEN DONE
 5603  016432                                 ENDSEG
  (3)  016432                          10000$:
```

```
 (3)  016432  104405                        TRAP    C$ESEG
5604  016434  012705  000001                MOV     #1,R5           ;START WITH BIT 0
5605                                ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
5606  016440                                BGNSEG
 (3)  016440  104404                        TRAP    C$BSEG
5607  016442                          67$:
5608  016442  005105                        COM     R5
5609  016444  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5610  016450                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  016450  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5611  016454  122102                        122100!2                ;MOV DATA TO IBUS* REGISTER 2
5612  016456                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  016456  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5613  016462  021045                        21005!<2*20>            ;READ FROM IBUS* REGISTER 2
5614  016464  116104  000005                MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
5615  016470  120504                        CMPB    R5,R4           ;DATA CORRECT?
5616  016472  001414                        BEQ     68$             ;BR IF YES
5617  016474                                BERROR  29              ;ERROR
 (5)  016514  104455                        TRAP    C$ERDF
 (6)  016516  000035                        .WORD   29
 (6)  016520  005605                        .WORD   EM29
 (6)  016522  010004                        .WORD   ERR29
5618  016524                          68$:  ESCAPE  SEG
 (3)  016524  104410                        TRAP    C$ESCAPE
 (3)  016526  000012                        .WORD   10001$-.
5619  016530  005105                        COM     R5              ;CHANGE TO FLOATING 1
5620  016532  000241                        CLC                     ;CLEAR CARRY
5621  016534  106105                        ROLB    R5              ;SHIFT BIT IN R5
5622  016536  001341                        BNE     67$             ;IF R5=0 THEN DONE
5623  016540                                ENDSEG
 (3)  016540                          10001$:
 (3)  016540  104405                        TRAP    C$ESEG
5624  016542                          ENDTST
 (3)  016542                          L10077:
 (3)  016542  104401                        TRAP    C$ETST
5625
5626  016544                                BADHEAD
 (2)                                    ;******************** TEST 23 ********************
5627                                    ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
5628                                    ;*FLOAT A 1 THOUGH IBUS REGISTER 3
5629                                    ;*FLOAT A 0 THROUGH IBUS REGISTER 3
5630  016544                                BADHEAD
 (2)                                    ;******************** TEST 23 ********************
5631
5632  016544                          BGNTST
 (3)  016544                          T23::
5633  016544                                MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  016544  004537  003156                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
5634  016550  012737  000003  002624        MOV     #3,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
5635  016556  012705  000001                MOV     #1,R5           ;START WITH BIT 0
5636  016562                                MYINT
 (1)  016562  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5637  016566                                BGNSEG
 (3)  016566  104404                        TRAP    C$BSEG
5638  016570                          64$:
5639  016570  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
```

```
5640  016574                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  016574  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5641  016600  122103                            122100.3                ;MOV DATA TO IBUS* REGISTER 3
5642  016602                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  016602  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5643  016606  021065                            21005!<3*20>            ;READ FROM IBUS* REGISTER 3
5644  016610  116104  000005                    MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
5645  016614  120504                            CMPB    R5,R4           ;DATA CORRECT?
5646  016616  001414                            BEQ     65$             ;BR IF YES
5647  016620                                    BERROR  29              ;ERROR
 (5)  016640  104455                            TRAP    C$ERDF
 (6)  016642  000035                            .WORD   29
 (6)  016644  005605                            .WORD   EM29
 (6)  016646  010004                            .WORD   ERR29
5648  016650                            65$:    ESCAPE  SEG
 (3)  016650  104410                            TRAP    C$ESCAPE
 (3)  016652  000010                            .WORD   10000$-.
5649  016654  000241                            CLC                     ;CLEAR CARRY
5650  016656  106105                            ROLB    R5              ;SHIFT BIT IN R5
5651  016660  001343                            BNE     64$             ;IF R5=0 THEN DONE
5652  016662                                    ENDSEG
 (3)  016662                            10000$:
 (3)  016662  104405                            TRAP    C$ESEG
5653  016664  012705  000000                    MOV     #1,R5           ;START WITH BIT 0
5654                                    ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
5655  016670                                    BGNSEG
 (3)  016670  104404                            TRAP    C$BSEG
5656  016672                            67$:
5657  016672  005105                            COM     R5
5658  016674  010561  000004                    MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5659  016700                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  016700  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5660  016704  122103                            122100!3                ;MOV DATA TO IBUS* REGISTER 3
5661  016706                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  016706  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5662  016712  021065                            21005!<3*20>            ;READ FROM IBUS* REGISTER 3
5663  016714  116104  000005                    MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
5664  016720  120504                            CMPB    R5,R4           ;DATA CORRECT?
5665  016722  001414                            BEQ     68$             ;BR IF YES
5666  016724                                    BERROR  29              ;ERROR
 (5)  016744  104455                            TRAP    C$ERDF
 (6)  015746  000035                            .WORD   29
 (6)  016750  005605                            .WORD   EM29
 (6)  016752  010004                            .WORD   ERR29
5667  016754                            68$:    ESCAPE  SEG
 (3)  016754  104410                            TRAP    C$ESCAPE
 (3)  016756  000012                            .WORD   10001$-.
5668  016760  005105                            COM     R5              ;CHANGE TO FLOATING 1
5669  016762  000241                            CLC                     ;CLEAR CARRY
5670  016764  106105                            ROLB    R5              ;SHIFT BIT IN R5
5671  016766  001341                            BNE     67$             ;IF R5=0 THEN DONE
5672  016770                                    ENDSEG
 (3)  016770                            10001$:
 (3)  016770  104405                            TRAP    C$ESEG
5673  016772                            ENDTST
 (3)  016772                            L10100:
```

```
 (3)  016772  104401                          TRAP    C$ETST
5674
5675  016774                                 BADHEAD
 (2)                                         ;****************** TEST 24 ******************
5676                                         ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
5677                                         ;*FLOAT A 1 THOUGH IBUS REGISTER 4
5678                                         ;*FLOAT A 0 THROUGH IBUS REGISTER 4
5679  016774                                 BADHEAD
 (2)                                         ;****************** TEST 24 ******************
5680
5681  016774                          BGNTST
 (3)  016774                          T24::
5682  016774                                 MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  016774  004537  003156                 JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
5683  017000  012737  000004  002624         MOV     #4,MR0          ;SAVE REGISTEF ADDRESS FOR TYPEOUT
5684  017006  012705  000001                MOV     #1,R5           ;START WITH BIT 0
5685  017012                                 MYINT
 (1)  017012  013701  002716                 MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5686  017016                                 BGNSEG
 (3)  017016  104404                         TRAP    C$BSEG
5687  017020                          64$:
5688  017020  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5689  017024                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  017024  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5690  017030  122104                        122100!4                ;MOV DATA TO IBUS* REGISTER 4
5691  017032                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  017032  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5692  017036  021105                        21005!<4*20>            ;READ FROM IBUS* REGISTER 4
5693  017040  116104  000005                MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
5694  017044  120504                        CMPB    R5,R4           ;DATA CORRECT?
5695  017046  001414                        BEQ     65$             ;BR IF YES
5696  017050                                 BERROR  29              ;ERROR
 (5)  017070  104455                         TRAP    C$ERDF
 (6)  017072  000035                        .WORD   29
 (6)  017074  005605                        .WORD   EM29
 (6)  017076  010004                        .WORD   ERR29
5697  017100                          65$:   ESCAPE  SEG
 (3)  017100  104410                         TRAP    C$ESCAPE
 (3)  017102  000010                        .WORD   10000$-.
5698  017104  000241                        CLC                     ;CLEAR CARRY
5699  017106  106105                        ROLB    R5              ;SHIFT BIT IN R5
5700  017110  001343                        BNE     64$             ;IF R5=0 THEN DONE
5701  017112                                 ENDSEG
 (3)  017112                          10000$:
 (3)  017112  104405                         TRAP    C$ESEG
5702  017114  012705  000001                MOV     #1,R5           ;START WITH BIT 0
5703                                 ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
5704  017120                                 BGNSEG
 (3)  017120  104404                         TRAP    C$BSEG
5705  017122                          67$:
5706  017122  005105                        COM     R5
5707  017124  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5708  017130                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  017130  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5709  017134  122104                        122100.4                ;MOV DATA TO IBUS* REGISTER 4
5710  017136                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
```

```
  (1)  017136  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5711   017142  021105                            21005!<4*20>            ;READ FROM IBUS* REGISTER 4
5712   017144  116104  000005                    MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
5713   017150  120504                            CMPB    R5,R4           ;DATA CORRECT?
5714   017152  001414                            BEQ     68$             ;BR IF YES
5715   017154                                    BERROR  29              ;ERROR
  (5)  017174  104455                            TRAP    C$ERDF
  (6)  017176  000035                            .WORD   29
  (6)  017200  005605                            .WORD   EM29
  (6)  017202  010004                            .WORD   ERR29
5716   017204                            68$:    ESCAPE  SEG
  (3)  017204  104410                            TRAP    C$ESCAPF
  (3)  017206  000012                            .WORD   10001$-.
5717   017210  005105                            COM     R5              ;CHANGE TO FLOATING 1
5718   017212  000241                            CLC                     ;CLEAR CARRY
5719   017214  106105                            ROLB    R5              ;SHIFT BIT IN R5
5720   017216  001341                            BNE     67$             ;IF R5=0 THEN DONE
5721   017220                                    ENDSEG
  (3)  017220                            10001$:
  (3)  017220  104405                            TRAP    C$ESEG
5722   017222                            ENDTST
  (3)  017222                            L1010':
  (3)  017222  104401                            TRAP    C$ETST
5723
5724   017224                                    BADHEAD
  (2)                                             ;****************** TEST 25 ******************
5725                                              ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
5726                                              ;*FLOAT A 1 THOUGH IBUS REGISTER 5
5727                                              ;*FLOAT A 0 THROUGH IBUS REGISTER 5
5728   017224                                     BADHEAD
  (2)                                             ;****************** TEST 25 ******************
5729
5730   017224                            BGNTST
  (3)  017224                            T25::
5731   017224                                    MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  017224  004537  003156                    JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
5732   017230  012737  000005  002624            MOV     #5,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
5733   017236  012705  000001                    MOV     #1,R5           ;START WITH BIT 0
5734   017242                                    MYINT
  (1)  017242  013701  002716                    MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5735   017246                                    BGNSEG
  (3)  017246  104404                            TRAP    C$BSEG
5736   017250                            64$:
5737   017250  010561  000004                    MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5738   017254                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  017254  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5739   017260  122105                            122100.5                ;MOV DATA TO IBUS* REGISTER 5
5740   017262                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  017262  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5741   017266  021125                            21005!<5*20>            ;READ FROM IBUS* REGISTER 5
5742   017270  116104  000005                    MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
5743   017274  120504                            CMPB    R5,R4           ;DATA CORRECT?
5744   017276  001414                            BEQ     65$             ;BR IF YES
5745   017300                                    BERROR  29              ;ERROR
  (5)  017320  104455                            TRAP    C$ERDF
  (6)  017322  000035                            .WORD   29
```

```
  (6)  017324  005605                        .WORD   EM29
  (6)  017326  010004                        .WORD   ERR29
5746   017330                        65$:    ESCAPE  SEG
  (3)  017330  104410                        TRAP    C$ESCAPE
  (3)  017332  000010                        .WORD   10000$-.
5747   017334  000241                        CLC                     ;CLEAR CARRY
5748   017336  106105                        ROLB    R5              ;SHIFT BIT IN R5
5749   017340  001343                        BNE     64$             ;IF R5=0 THEN DONE
5750   017342                                ENDSEG
  (3)  017342                        10000$:
  (3)  017342  104405                        TRAP    C$ESEG
5751   017344  012705  000001                MOV     #1,R5           ;START WITH BIT 0
5752                                ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
5753   017350                                BGNSEG
  (3)  017350  104404                        TRAP    C$BSEG
5754   017352                        67$:
5755   017352  005105                        COM     R5
5756   017354  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5757   017360                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  017360  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5758   017364  122105                        122100.5                ;MOV DATA TO IBUS* REGISTER 5
5759   017366                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  017366  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5760   017372  021125                        21005!<5*20>            ;READ FROM IBUS* REGISTER 5
5761   017374  116104  000005                MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
5762   017400  120504                        CMPB    R5,R4           ;DATA CORRECT?
5763   017402  001414                        BEQ     68$             ;BR IF YES
5764   017404                                BERROR  29              ;ERROR
  (5)  017424  104455                        TRAP    C$ERDF
  (6)  017426  000035                        .WORD   29
  (6)  017430  005605                        .WORD   EM29
  (6)  017432  010004                        .WORD   ERR29
5765   017434                        68$:    ESCAPE  SEG
  (3)  017434  104410                        TRAP    C$ESCAPE
  (3)  017436  000012                        .WORD   10001$-.
5766   017440  005105                        COM     R5              ;CHANGE TO FLOATING 1
5767   017442  000241                        CLC                     ;CLEAR CARRY
5768   017444  106105                        ROLB    R5              ;SHIFT BIT IN R5
5769   017446  001341                        BNE     67$             ;IF R5-0 THEN DONE
5770   017450                                ENDSEG
  (3)  017450                        10001$:
  (3)  017450  104405                        TRAP    C$ESEG
5771   017452                        ENDTST
  (3)  017452                        L10102:
  (3)  017452  104401                        TRAP    C$ETST
5772
5773   017454                                BADHEAD
  (2)                                       ;******************** TEST 26 ********************
5774                                        ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
5775                                        ;*FLOAT A 1 THOUGH IBUS REGISTER 6
5776                                        ;*FLOAT A 0 THROUGH IBUS REGISTER 6
5777   017454                               BADHEAD
  (2)                                       ;******************** TEST 26 ********************
5778
5779   017454                        BGNTST
  (3)  017454                        T26::
```

```
 5780  017454                                 MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  017454  004537  003156                 JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 5781  017460  012737  000006  002624         MOV     #6,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
 5782  017466  012705  000001                 MOV     #1,R5           ;START WITH BIT 0
 5783  017472                                 MYINT
  (1)  017472  013701  002716                 MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 5784  017476                                 BGNSEG
  (3)  017476  104404                         TRAP    C$BSEG
 5785  017500                          64$:
 5786  017500  010561  000004                 MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 5787  017504                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  017504  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 5788  017510  122106                         122100!6                ;MOV DATA TO IBUS* REGISTER 6
 5789  017512                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  017512  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 5790  017516  021145                         21005!<6*20>            ;READ FROM IBUS* REGISTER 6
 5791  017520  116104  000005                 MOVB    5(R1),R4        ;PUT 'FOUND'' INTO R4
 5792  017524  120504                         CMPB    R5,R4           ;DATA CORRECT?
 5793  017526  001414                         BEQ     65$             ;BR IF YES
 5794  017530                                 @ERROR  29              ;ERROR
  (5)  017550  104455                         TRAP    C$ERDF
  (6)  017552  000035                         .WORD   29
  (6)  017554  005605                         .WORD   EM29
  (6)  017556  010004                         .WORD   ERR29
 5795  017560                          65$:   ESCAPE  SEG
  (3)  017560  104410                         TRAP    C$ESCAPE
  (3)  017562  000010                         .WORD   10000$-.
 5796  017564  000241                         CLC                     ;CLEAR CARRY
 5797  017566  106105                         ROLB    R5              ;SHIFT BIT IN R5
 5798  017570  001343                         BNE     64$             ;IF R5=0 THEN DONE
 5799  017572                                 ENDSEG
  (3)  017572                          10000$:
  (3)  017572  104405                         TRAP    C$ESEG
 5800  017574  012705  000001                 MOV     #1,R5           ;START WITH BIT 0
 5801                                 ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
 5802  017600                                 BGNSEG
  (3)  017600  104404                         TRAP    C$BSEG
 5803  017602                          67$:
 5804  017602  005105                         COM     R5
 5805  017604  010561  000004                 MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 5806  017610                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  017610  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 5807  017614  122106                         122100!6                ;MOV DATA TO IBUS* REGISTER 6
 5808  017616                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  017616  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 5809  017622  021145                         21005!<6*20>            ;READ FROM IBUS* REGISTER 6
 5810  017624  116104  000005                 MOVB    5(R1),R4        ;PUT 'FOUND'' INTO R4
 5811  017630  120504                         CMPB    R5,R4           ;DATA CORRECT?
 5812  017632  001414                         BEQ     68$             ;BR IF YES
 5813  017634                                 BERROR  29              ;ERROR
  (5)  017654  104455                         TRAP    C$ERDF
  (6)  017656  000035                         .WORD   29
  (6)  017660  005605                         .WORD   EM29
  (6)  017662  010004                         .WORD   ERR29
 5814  017664                          68$:   ESCAPE  SEG
  (3)  017664  104410                         TRAP    C$ESCAPE
```

```
 (3)  017666  000012                        .WORD   10001$-.
5815  017670  005105                        COM     R5              ;CHANGE TO FLOATING 1
5816  017672  000241                        CLC                     ;CLEAR CARRY
5817  017674  106105                        ROLB    R5              ;SHIFT BIT IN R5
5818  017676  001341                        BNE     67$             ;IF R5=0 THEN DONE
5819  017700                                ENDSEG
 (3)  017700                          10001$:
 (3)  017700  104405                        TRAP    C$ESEG
5820  017702                          ENDTST
 (3)  017702                          L10103:
 (3)  017702  104401                        TRAP    C$ETST
5821
5822  017704                                BADHEAD
 (2)                                        ;****************** TEST 27 ******************
5823                                        ;*MICRO PROCEOR IBUS* REGISTER WRITE/READ TEST
5824                                        ;*FLOAT A 1 THOUGH IBUS* REGISTER 7
5825                                        ;*FLOAT A 0 THROUGH IBUS* REGISTER 7
5826  017704                                BADHEAD
 (2)                                        ;****************** TEST 27 ******************
5827
5828  017704                          BGNTST
 (3)  017704                          T27::
5829  017704                                MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  017704  004537  003156                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
5830  017710  012737  000007  002624        MOV     #7,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
5831  017716  012705  000001                MOV     #1,R5           ;START WITH BIT 0
5832  017722                                MYINT
 (1)  017722  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5833  017726                                BGNSEG
 (3)  017726  104404                        TRAP    C$BSEG
5834  017730                          64$:
5835  017730  010561  000004                MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5836  017734                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  017734  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5837  017740  122107                        122100!7                ;MOV DATA TO IBUS* REGISTER 7
5838  017742                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  017742  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
5839  017746  021165                        21005!<7*20>            ;READ FROM IBUS* REGISTER 7
5840  017750  116104  000005                MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
5841  017754  120504                        CMPB    R5,R4           ;DATA CORRECT?
5842  017756  001414                        BEQ     65$             ;BR IF YES
5843  017760                                BERROR  29              ;ERROR
 (5)  020000  104455                        TRAP    C$ERDF
 (6)  020002  000035                        .WORD   29
 (6)  020004  005605                        .WORD   EM29
 (6)  020006  010004                        .WORD   ERR29
5844  020010                          65$:  ESCAPE  SEG
 (3)  020010  104410                        TRAP    C$ESCAPE
 (3)  020012  000010                        .WORD   10000$-.
5845  020014  000241                        CLC                     ;CLEAR CARRY
5846  020016  106105                        ROLB    R5              ;SHIFT BIT IN R5
5847  020020  001343                        BNE     64$             ;IF R5-0 THEN DONE
5848  020022                                ENDSEG
 (3)  020022                          10000$:
 (3)  020022  104405                        TRAP    C$ESEG
5849  020024  012705  000001                MOV     #1,R5           ;START WITH BIT 0
```

```
5850
5851  020030
 (3)  020030  104404                    ;69$:   COM     R5              ;CHANGE TO FLOATING ZERO
                                               BGNSEG
                                               TRAP    C$BSEG
5852  020032                           67$:
5853  020032  005105                           COM     R5
5854  020034  010561  000004                   MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
5855  020040                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  020040  004537  003244                   JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5856  020044  122107                           122100!7                ;MOV DATA TO IBUS* REGISTER 7
5857  020046                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  020046  004537  003244                   JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5858  020052  021165                           21005!<7*20>            ;READ FROM IBUS* REGISTER 7
5859  020054  116104  000005                   MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
5860  020060  120504                           CMPB    R5,R4           ;DATA CORRECT?
5861  020062  001414                           BEQ     68$             ;BR IF YES
5862  020064                                   BERROR  29              ;ERROR
 (5)  020104  104455                           TRAP    C$ERDF
 (6)  020106  000035                           .WORD   29
 (6)  020110  005605                           .WORD   EM29
 (6)  020112  010004                           .WORD   ERR29
5863  020114                           68$:    ESCAPE  SEG
 (3)  020114  104410                           TRAP    C$ESCAPE
 (3)  020116  000012                           .WORD   10001$-.
5864  020120  005105                           COM     R5              ;CHANGE TO FLOATING 1
5865  020122  000241                           CLC                     ;CLEAR CARRY
5866  020124  106105                           ROLB    R5              ;SHIFT BIT IN R5
5867  020126  001341                           BNE     67$             ;IF R5=0 THEN DONE
5868  020130                                   ENDSEG
 (3)  020130                           10001$:
 (3)  020130  104405                           TRAP    C$ESEG
5869  020132                           ENDTST
 (3)  020132                           L10104:
 (3)  020132  104401                           TRAP    C$ETST
5870
5871  020134                                   BADHEAD
 (2)                                             ;********** TEST 28 **********
5872                                             ;*MICRO PROCESSOR IBUS DUAL ADDRESS TEST
5873                                             ;*WRITE ALL IBUS REGISTERS WITH INCREMENTING PATTERN
5874                                             ;*READ ALL IBUS REGISTERS TO VERIFY CORRECT ADDRESSING
5875  020134                                   BADHEAD
 (2)                                             ;********** TEST 28 **********
5876
5877  020134                           BGNTST
 (3)  020134                           T28::
5878  020134                                   MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  020134  004537  003156                   JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
5879  020140  012705  000001                   MOV     #1,R5           ;START WITH A ONE
5880  020144  005002                           CLR     R2              ;R2 CONTAINS ADDRESS OF REGISTER
5881  020146                                   MYINT
 (1)  020146  013701  002716                   MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
5882  020152                                   BGNSEG
 (3)  020152  104404                           TRAP    C$BSEG
5883  020154  010203                   1$:     MOV     R2,R3           ;R3=REGISTER ADDRESS
5884  020156  010561  000004                   MOV     R5,4(R1)        ;WRITE DATA TO PORT4
5885  020162  042737  000017  020200         BIC     #17,5$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
5886  020170  050337  020200                   BIS     R3,5$           ;ADD ADDRESS TO INSTRUCTION
```

```
5887  020174                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  020174  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5888  020200  122100                        5$:     122100                  ;MOVE DATA TO IBUS REGISTER
5889  020202  006303                                ASL     R3              ;SHIFT ADDRESS
5890  020204  006303                                ASL     R3              ;4 TIMES TO GET
5891  020206  006303                                ASL     R3              ;IT TO BITS 4-7
5892  020210  006303                                ASL     R3              ;OF NEXT INSTRUCTION
5893  020212  042737  000360  020230                BIC     #360,6$         ;CLEAR ADDRESS FIELD
5894  020220  050337  020230                        BIS     R3,6$           ;ADD ADDRESS TO INSTRUCTION
5895  020224                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  020224  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5896  020230  021005                        6$:     21005                   ;READ FROM IBUS REGISTER
5897  020232  116104  000005                        MOVB    5(R1),R4        ;PUT 'FOUND' IN R4
5898  020236  120504                                CMPB    R5,R4           ;IS DATA CORRECT?
5899  020240  001414                                BEQ     2$              ;BR IF YES
5900  020242                                        BERROR  29              ;DATA ERROR
 (5)  020262  104455                                TRAP    C$ERDF
 (6)  020264  000035                                .WORD   29
 (6)  020266  005605                                .WORD   EM29
 (6)  020270  010004                                .WORD   ERR29
5901  020272                                2$:     ESCAPE  SEG
 (3)  020272  104410                                TRAP    C$ESCAPE
 (3)  020274  000014                                .WORD   10000$-.
5902  020276  005205                                INC     R5              ;INCREMENT PATTERN
5903  020300  005202                                INC     R2              ;INCREMENT REGISTER ADDRESS
5904  020302  022702  000010                        CMP     #7+1,R2         ;LAST ADDRESS DONE?
5905  020306  001322                                BNE     1$              ;BR IF NO
5906  020310                                        ENDSEG
 (3)  020310                                10000$:
 (3)  020310  104405                                TRAP    C$ESEG
5907  020312  012705  000001                        MOV     #1,R5           ;RESTART PATTERN TO 1
5908  020316  005002                                CLR     R2              ;RESTART AT ADDRESS 0
5909  020320                                        BGNSEG
 (3)  020320  104404                                TRAP    C$BSEG
5910  020322  005003                                CLR     R3              ;RESTART AT ADDRESS 0
5911  020324  042737  000360  020342        3$:     BIC     #360,7$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
5912  020332  050337  020342                        BIS     R3,7$           ;ADD ADDRESS TO INSTRUCTION
5913  020336                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  020336  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
5914  020342  021005                        7$:     21005                   ;READ FROM IBUS REGISTER
5915  020344  116104  000005                        MOVB    5(R1),R4        ;PUT 'FOUND' IN $GDDAT
5916  020350  120504                                CMPB    R5,R4           ;DATA CORRECT?
5917  020352  001414                                BEQ     4$              ;BR IF YES
5918  020354                                        BERROR  30              ;DUAL ADDRESSING ERROR
 (5)  020374  104455                                TRAP    C$ERDF
 (6)  020376  000036                                .WORD   30
 (6)  020400  004353                                .WORD   EM30
 (6)  020402  010066                                .WORD   ERR30
5919  020404                                4$:     ESCAPE  SEG
 (3)  020404  104410                                TRAP    C$ESCAPE
 (3)  020406  000020                                .WORD   10001$-.
5920  020410  005205                                INC     R5              ;INCREMENT PATTERN
5921  020412  005202                                INC     R2              ;NEXT ADDRESS
5922  020414  062703  000020                        ADD     #20,R3          ;ADD 1 TO ADDRESS IN R3(SHIFTED 4 TIMES)
5923  020420  022702  000010                        CMP     #7+1,R2         ;LAST ADDRESS DONE?
5924  020424  001357                                BNE     3$              ;BR IF NO
```

```
5925  020426                                          ENDSEG
 (3)  020426                                  10001$:
 (3)  020426  104405                                 TRAP    C$ESEG
5926  020430                                  ENDTST
 (3)  020430                                  L10105:
 (3)  020430  104401                                 TRAP    C$ETST
5927
5928  020432                                        BADHEAD
 (2)                                                ;****************** TEST 29 ******************
5929                                                ;*MICRO PROCESSOR BR REGISTER TEST
5930                                                ;*FLOAT A 1 THOUGH THE BR
5931                                                ;*FLOAT A 0 THOUGH THE BR
5932  020432                                        BADHEAD
 (2)                                                ;****************** TEST 29 ******************
5933
5934  020432                                  BGNTST
 (3)  020432                                  T29::
5935                                                                    ;R1 CONTAINS BASE M8200,4,7 ADDRESS
5936  020432                                        MSTCLR            ;MASTER CLEAR COMM. MICRO-PROCCESSOR FAMILY
 (1)  020432  004537  003156                         JSR     R5,.MSTCLR          ;CLEAR M8200,4,7
5937  020436  012702  000001                         MOV     #1,R2     ;START PATTERN WITH BIT0
5938  020442                                        MYINT
 (1)  020442  013701  002716                         MOV     KMCSR,R1  ;GET DEVICE ADDRESS.
5939  020446                                        BGNSEG
 (3)  020446  104404                                 TRAP    C$BSEG
5940  020450                                  64$:
5941  020450  010261  000004                         MOV     R2,4(R1)  ;WRITE PATTERN IN PORT4
5942  020454                                        ROMCLK              ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  020454  004537  003244                         JSR     R5,.ROMCLK          ;CLOCK INSTRUCTION
5943  020460  120500                               120500              ;MOVE DATA TO THE BR REGISTER
5944  020462                                        ROMCLK              ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  020462  004537  003244                         JSR     R5,.ROMCLK          ;CLOCK INSTRUCTION
5945  020466  061225                               061225              ;MOVE BR TO PORT 5
5946  020470  116104  000005                         MOVB    5(R1),R4  ;PUT 'FOUND' IN R4
5947  020474  120204                               CMPB    R2,R4       ;IS DATA CORRECT?
5948  020476  001414                               BEQ     65$         ;BR IF YES
5949  020500                                        BERROR  3           ;DATA ERROR
 (5)  020520  104455                               TRAP    C$ERDF
 (6)  020522  000003                               .WORD   3
 (6)  020524  004416                               .WORD   EM3
 (6)  020526  006210                               .WORD   ERR3
5950  020530                                  65$:  ESCAPE  SEG
 (3)  020530  104410                               TRAP    C$ESCAPE
 (3)  020532  000010                               .WORD   10000$-.
5951  020534  000241                               CLC                 ;CLEAR CARRY
5952  020536  106102                               ROLB    R2          ;SHIFT BIT IN R2
5953  020540  001343                               BNE     64$         ;DONE IF R2-0
5954  020542                                        ENDSEG
 (3)  020542                                  10000$:
 (3)  020542  104405                               TRAP    C$ESEG
5955  020544  012702  000001                         MOV     #1,R2     ;START PATTERN WITH BIT0
5956  020550                                  69$:
5957  020550                                        BGNSEG
 (3)  020550  104404                               TRAP    C$BSEG
5958  020552                                  67$:
5959  020552  005102                               COM     R2
```

```
 5960  020554  010261  000004                MOV     R2,4(R1)        ;WRITE PATTERN IN PORT4
 5961  020560                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (*)  020560  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5962  020564  120500                        120500                  ;MOVE DATA TO THE BR REGISTER
 5963  020566                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  020566  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5964  020572  061225                        061225                  ;MOVE DR TO PORT 5
 5965  020574  116104  000005                MOVB    5(R1),R4        ;PUT 'FOUND' IN $GDDAT
 5966  020600  010205                        MOV     R2,R5
 5967  020602  120204                        CMPB    R2,R4           ;DATA CORRECT?
 5968  020604  001414                        BEQ     68$             ;BR IF YES
 5969  020606                                BERROR  3               ;DATA ERROR
  (5)  020626  104455                        TRAP    C$ERDF
  (6)  020630  000003                        .WORD   3
  (5)  020632  004416                        .WORD   EM3
  (6)  020634  006210                        .WORD   ERR3
 5970  020636                        68$:    ESCAPE  SEG
  (3)  020636  104410                        TRAP    C$ESCAPE
  (3)  020640  000016                        .WORD   10001$-.
 5971                                                                ;FAILED TO CLEAR
 5972  020642  105061  000001        70$:    CLRB    1(R1)           ;BRG
 5973  020646  005102                        COM     R2              ;CHANGE BACK TO A ONE
 5974  020650  000241                        CLC                     ;CLEAR CARRY
 5975  020652  106102                        ROLB    R2              ;SHIFT BIT IN R5
 5976  020654  001336                        BNE     67$             ;DONE IF R5-0
 5977  020656                                ENDSEG
  (3)  020656                        10001$:
  (3)  020656  104405                        TRAP    C$ESEG
 5978  020660                        ENDTST
  (3)  020660                        L10106:
  (3)  020660  104401                        TRAP    C$ETST
 5979
 5980  020662                                BADHEAD
  (2)                                       ;********************* TEST 30 *********************
 5981                                       ;*SCRATCH PAD TEST
 5982                                       ;*FLOAT A 1 THOUGH EACH SCRATCH PAD LOCATION
 5983                                       ;*FLOAT A 0 THOUGH EACH SCRATCH PAD LOCATION
 5984  020662                                BADHEAD
  (2)                                       ;********************* TEST 30 *********************
 5985
 5986  020662                        BGNTST
  (3)  020662                        T30::
 5987  020662                                MYINT
  (1)  020662  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 5988  020666                                MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  020666  004537  003156                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 5989  020672  005002                        CLR     R2              ;START AT ADDRESS ZERO
 5990  020674  012705  000001                MOV     #1,R5           ;START WITH BIT0
 5991  020700                                BGNSUB
  (3)  020700                        T30.1:
  (3)  020700  104402                        TRAP    C$BSUB
 5992  020702                        1$:     BGNSEG
  (3)  020702  104404                        TRAP    C$BSEG
 5993  020704  042737  000017  020726  64$:  BIC     #17,65$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
 5994  020712  050237  020726                BIS     R2,65$          ;ADD ADDRESS TO INSTRUCTION
 5995  020716  010561  000004                MOV     R5,4(R1)        ;WRITE PATTERN IN PORT4
```

```
 5996  020722                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  020722  004537  003244                       JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 5997  020726  123100                        65$:    123100                  ;WRITE SCRATCH PAD(ADDRESS IN R2)
 5998  020730  042737  000017  020746              BIC     #17,66$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
 5999  020736  050237  020746                      BIS     R2,66$          ;ADD ADDRESS TO INSTRUCTION
 6000  020742                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  020742  004537  003244                       JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6001  020746  040600                        66$:    040600                  ;MOVE SP TO BR
 6002  020750                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  020750  004537  003244                       JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6003  020754  061225                              061225                  ;MOVE BR TO PORT5
 6004  020756  010537  002636                      MOV     R5,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
 6005  020762  116104  000005                      MOVB    5(R1),R4        ;PUT "FOUND" IN R4
 6006  020766  123704  002636                      CMPB    $GDDAT,R4               ;DATA CORRECT
 6007  020772  001414                              BEQ     67$             ;BR IF YES
 6008  020774                                        RERROR  4               ;DATA ERROR
  (5)  021014  104455                              TRAP    C$ERDF
  (6)  021016  000004                              .WORD   4
  (6)  021020  004444                              .WORD   EM4
  (6)  021022  006266                              .WORD   ERR4
 6009  021024                                67$:    ESCAPE  SEG
  (3)  021024  104410                              TRAP    C$ESCAPE
  (3)  021026  000010                              .WORD   10000$-.
 6010  021030  000241                              CLC                     ;CLEAR CARRY
 6011  021032  106105                              ROLB    R5              ;SHIFT BIT IN R5
 6012  021034  001323                              BNE     64$             ;DONE IF R5=0
 6013  021036                                        ENDSEG
  (3)  021036                                10000$:
  (3)  021036  104405                              TRAP    C$ESEG
 6014  021040  012705  000001                      MOV     #1,R5           ;START WITH BIT0
 6015  021044                                        BGNSEG
  (3)  021044  104404                              TRAP    C$BSEG
 6016
 6017  021046  005105                        73$:    COM     R5              ;CHANGE TO FLOATING ZERO
 6018  021050  042737  000017  021072      69$:    BIC     #17,70$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
 6019  021056  050237  021072                      BIS     R2,70$          ;ADD ADDRESS TO INSTRUCTION
 6020  021062  010561  000004                      MOV     R5,4(R1)        ;WRITE PATTERN IN PORT4
 6021  021066                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  021066  004537  003244                       JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6022  021072  123100                        70$:    123100                  ;WRITE SCRATCH PAD(ADDRESS IN R2)
 6023  021074  042737  000017  021112              BIC     #17,71$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
 6024  021102  050237  021112                      BIS     R2,71$          ;ADD ADDRESS TO INSTRUCTION
 6025  021106                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  021106  004537  003244                       JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6026  021112  040600                        71$:    040600                  ;MOVE SP TO BR
 6027  021114                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  021114  004537  003244                       JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6028  021120  061225                              061225                  ;MOVE BR TO PORT5
 6029  021122  010537  002636                      MOV     R5,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
 6030  021126  116104  000005                      MOVB    5(R1),R4        ;PUT "FOUND" IN $GDDAT
 6031  021132  123704  002636                      CMPB    $GDDAT,R4               ;DATA CORRECT?
 6032  021136  001414                              BEQ     72$             ;BR IF YES
 6033  021140                                        RERROR  4               ;DATA ERROR
  (5)  021160  104455                              TRAP    C$ERDF
  (6)  021162  000004                              .WORD   4
  (6)  021164  004444                              .WORD   EM4
```

```
  (6)  021166  006266                        .WORD   ERR4
 6034  021170                        72$:    ESCAPE  TST
  (3)  021170  104410                        TRAP    C$ESCAPE
  (3)  021172  000032                        .WORD   L10107-.
 6035  021174  005105                        COM     R5              ;CHANGE BACK TO A ONE
 6036  021176  000241                        CLC                     ;CLEAR CARRY
 6037  021200  106105                        ROLB    R5              ;SHIFT BIT IN R5
 6038  021202  001321                        BNE     73$             ;DONE IF R5=0
 6039  021204                        ENDSEG
  (3)  021204                        10001$:
  (3)  021204  104405                        TRAP    C$ESEG
 6040  021206  012705  000001              MOV     #1,R5           ;RESTART AT BIT 0
 6041  021212  005202                        INC     R2              ;NEXT SP ADDRESS
 6042  021214  022702  000020              CMP     #20,R2          ;LAST ADDRESS?
 6043  021220  001230                        BNE     1$              ;BR IF NO
 6044  021222                              ENDSUB
  (3)  021222                        L10110:
  (3)  021222  104403                        TRAP    C$ESUB
 6045  021224                        ENDTST
  (3)  021224                        L10107:
  (3)  021224  104401                        TRAP    C$ETST
 6046
 6047  021226                              BADHEAD
  (2)                                       ;****************** TEST 31 ******************
 6048                                       ;*SCRATCH PAD DUAL ADDRESSING TEST
 6049                                       ;*WRITE AN INCREMENTING PATTERN IN ALL SP LOCATIONS
 6050                                       ;*READ ALL SP LOCATIONS TO VERIFY CORRECT ADDRESSING
 6051  021226                              BADHEAD
  (2)                                       ;****************** TEST 31 ******************
 6052
 6053  021226                        BGNTST
  (3)  021226                        T31::
 6054  021226                              MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  021226  004537  003156              JSR     R5,.MSTCLR              ;CLEAR MS200,4,7
 6055  021232  012705  000001              MOV     #1,R5           ;START WITH A 1
 6056  021236  005003                        CLR     R3              ;ADDRESS 0
 6057  021240                              MYINT
  (1)  021240  013701  002716              MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6058  021244                              BGNSEG
  (3)  021244  104404                        TRAP    C$BSEG
 6059  021246  010302                  1$:   MOV     R3,R2           ;MOVE ADDRESS TO R2
 6060  021250  042737  000017  021272      BIC     #17,2$          ;CLEAR ADDRESS FIELD
 6061  021256  050237  021272              BIS     R2,2$           ;ADD ADDRESS TO INSTRUCTION
 6062  021262  010561  000004              MOV     R5,4(R1)        ;WRITE PATTERN IN PORT4
 6063  021266                              ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  021266  004537  003244              JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6064  021272  123100                  2$:   123100                  ;WRITE SP(ADDRESS IN R2)
 6065  021274  042737  000017  021312      BIC     #17,3$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
```

```
 6067  021302  050237  021312                BIS     R2,3$           ;ADD ADDRESS TO INSTRUCTION
 6068  021306                                ROMCLK                  ;NEXT WORD IS INSTRUCTION.  BBN
  (*)  021306  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
```

```
6070  021312  060600                  3$:     60600                   ;MOVE SP TO BR
6071  021314                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  021314  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6072  021320  061225                          61225                   ;MOVE BR TO PORT5
6073  021322  010537  002636                  MOV     R5,$GDDAT               ;PUT 'EXPECTED' IN $GDDAT
6074  021326  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND' IN R4
6075  021332  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT
6076  021336  001414                          BEQ     4$              ;BR IF YES
6077  021340                                  RERROR  4               ;DATA ERROR
 (5)  021360  104455                          TRAP    C$ERDF
 (6)  021362  000004                          .WORD   4
 (6)  021364  004444                          .WORD   EM4
 (6)  021366  006266                          .WORD   ERR4
6078  021370                          4$:     ESCAPE  SEG
 (3)  021370  104410                          TRAP    C$ESCAPE
 (3)  021372  000014                          .WORD   10000$-.
6079  021374  005205                          INC     R5              ;INCREMENT PATTERN
6080  021376  005203                          INC     R3              ;NEXT ADDRESS
6081  021400  022703  000020                  CMP     #20,R3          ;LAST ADDRESS DONE?
6082  021404  001320                          BNE     1$              ;BR IF NO
6083  021406                                  ENDSEG
 (3)  021406                          10000$:
 (3)  021406  104405                          TRAP    C$ESEG
6084  021410  012705  000001                  MOV     #1,R5           ;RESTART PATTERN AT 1
6085  021414  005003                          CLR     R3              ;RESTART AT ADDRESS ZERO
6086  021416                                  BGNSEG
 (3)  021416  104404                          TRAP    C$BSEG
6087  021420  010302                  5$:     MOV     R3,R2           ;PUT ADDRESS IN R2
6088  021422  042737  000017  021440  69$:    BIC     #17,6$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
6089  021430  050237  021440                  BIS     R2,6$           ;ADD ADDRESS TO INSTRUCTION
6090  021434                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  021434  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6091  021440  060600                  6$:     60600                   ;MOV SP TO BR
6092  021442                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  021442  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6093  021446  061225                          61225                   ;MOV BR TO PORT5
6094  021450  010537  002636                  MOV     R5,$GDDAT               ;PUT 'EXPECTED' IN $GDDAT
6095  021454  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND' IN $GDDAT
6096  021460  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
6097  021464  001414                          BEQ     7$              ;BR IF YES
6098  021466                                  RERROR  5               ;SP ADDRESSING ERROR
 (5)  021506  104455                          TRAP    C$ERDF
 (6)  021510  000005                          .WORD   5
 (6)  021512  004472                          .WORD   EM5
 (6)  021514  006350                          .WORD   ERR5
6099  021516                          7$:     ESCAPE  SEG
 (3)  021516  104410                          TRAP    C$ESCAPE
 (3)  021520  000014                          .WORD   10001$-.
6100  021522  005205                          INC     R5              ;INCREMENT PATTERN
6101  021524  005203                          INC     R3              ;NEXT ADDRESS
6102  021526  022703  000020                  CMP     #20,R3          ;LAST ADDRESS DONE?
6103  021532  001332                          BNE     5$              ;BR IF NO
6104  021534                                  ENDSEG
 (3)  021534                          10001$:
 (3)  021534  104405                          TRAP    C$ESEG
6105  021536                          ENDTST
```

```
  (3)  021536                          L10111:
  (3)  021536  104401                          TRAP    C$ETST
 6106
 6107  021540                                  BADHEAD
  (2)                                         ;********************* TEST 32 *********************
 6108                                         ;*INTERRUPT TEST
 6109                                         ;*TEST THAT DEVICE CAN INTERRUPT TO VECTOR A
 6110  021540                                  BADHEAD
  (2)                                         ;********************* TEST 32 *********************
 6111
 6112  021540                          BGNTST
  (3)  021540                          T32::
 6113  021540                                  MYINT
  (1)  021540  013701  002716                 MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6114  021544                                  BRESET                  ;BUS RESET
  (3)  021544  104433                         TRAP    C$RESET
 6115  021546  005011                         CLR     (R1)            ;CLEAR RUN
 6116  021550  004537  003552                 JSR     R5,SETVEC       ;SET UP VECTORS
 6117  021554  021674                         3$                      ;XX0
 6118  021556  021646                         2$                      ;XX4
 6119  021560  000340  000340                 .WORD   340,340         ;LEVEL 7
 6120  021564                          1$:     SETPRI  #PRI07          ;PS = LEVEL 7
  (3)  021564  012700  000340                 MOV     #PRI07,R0
  (3)  021570  104441                         TRAP    C$SPRI
 6121  021572  012761  000200  000004         MOV     #200,4(R1)      ;WRITE PORT4
 6122  021600                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  021600  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6123  021604  121111                         121111                  ;SET BR RQ IN IBUS* REG 11
 6124  021606                                  SETPRI  #PRI00          ;ALLOW INTERRUPT
  (3)  021606  012700  000000                 MOV     #PRI00,R0
  (3)  021612  104441                         TRAP    C$SPRI
 6125  021614  000240                         NOP
 6126  021616                                  ERROR   31              ;NO INTERRUPT
  (5)  021634  104455                         TRAP    C$ERDF
  (6)  021636  000037                         .WORD   31
  (6)  021640  005312                         .WORD   EM31
  (6)  021642  010144                         .WORD   ERR31
 6127  021644  000415                         BR      4$
 6128  021646                          2$:     ERROR   32              ;WRONG VECTOR
  (5)  021664  104455                         TRAP    C$ERDF
  (6)  021666  000040                         .WORD   32
  (6)  021670  005341                         .WORD   EM32
  (6)  021672  010172                         .WORD   ERR32
 6129  021674  062706  000004          3$:     ADD     #4,SP   ;RESET STACK
 6130  021700                          4$:
 6131  021700                          ENDTST
  (3)  021700                          L10112:
  (3)  021700  104401                          TRAP    C$ETST
 6132
 6133  021702                                  BADHEAD
  (2)                                         ;********************* TEST 33 *********************
 6134                                         ;*INTERRUPT TEST
 6135                                         ;*TEST THAT DEVICE CAN INTERRUPT TO VECTOR B
 6136  021702                                  BADHEAD
  (2)                                         ;********************* TEST 33 *********************
 6137
```

```
 6138  021702                                BGNTST
  (3)  021702                                T33::
 6139  021702                                        MYINT
  (1)  021702  013701  002716                       MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6140  021706                                       MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  021706  004537  003156                       JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 6141  021712  004537  003552                       JSR     R5,SETVEC       ;SET UP VECTORS
 6142  021716  022010                               2$                      ;XX0
 6143  021720  022036                               3$                      ;XX4
 6144  021722  000340  000340                       .WORD   340,340         ;LEVEL 7
 6145  021726                                1$:     SETPRI  #PRI07          ;PS = LEVEL 7
  (3)  021726  012700  000340                       MOV     #PRI07,R0
  (3)  021732  104441                               TRAP    C$SPRI
 6146  021734  012761  000300  000004               MOV     #300,4(R1)      ;WRITE PORT4
 6147  021742                                       ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  021742  004537  003244                       JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6148  021746  121111                               121111                  ;SET BR RQ IN IBUS* REG 11
 6149  021750                                       SETPRI  #PRI00          ;ALLOW INTERRUPT
  (3)  021750  012700  000000                       MOV     #PRI00,R0
  (3)  021754  104441                               TRAP    C$SPRI
 6150  021756  000240                               NOP
 6151  021760                                       ERROR   31              ;NO INTERRUPT
  (5)  021776  104455                               TRAP    C$ERDF
  (6)  022000  000037                               .WORD   31
  (6)  022002  005312                               .WORD   EM31
  (6)  022004  010144                               .WORD   ERR31
 6152  022006  000415                               BR      4$
 6153  022010                                2$:     ERROR   32              ;WRONG VECTOR
  (5)  022026  104455                               TRAP    C$ERDF
  (6)  022030  000040                               .WORD   32
  (6)  022032  005341                               .WORD   EM32
  (6)  022034  010172                               .WORD   ERR32
 6154  022036  062706  000004            3$:     ADD     #4,SP   ;RESET STACK
 6155  022042                             4$:
 6156  022042                             ENDTST
  (3)  022042                             L10113:
  (3)  022042  104401                               TRAP    C$ETST
 6157
 6158  022044                                       BADHEAD
  (2)                                              ;******************** TEST 34 ********************
 6159                                              ;*PRIORITY INTERRUPT TEST
 6160                                              ;*SET PS TO ALL BR LEVELS EQUAL OR GREATER THAN
 6161                                              ;*THE M8200,4,7 LEVEL, VERIFY THAT M8200,4,7 DOES NOT INTERRUPT
 6162  022044                                       BADHEAD
  (2)                                              ;******************** TEST 34 ********************
 6163
 6164  022044                             BGNTST
  (3)  022044                             T34::
 6165  022044                                        MYINT
  (1)  022044  013701  002716                       MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6166  022050                                       MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  022050  004537  003156                       JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 6167  022054  012704  000340                       MOV     #340,R4         ;PUT LEVEL 7 IN R2
 6168  022060                                       SETPRI  R4              ;SET PRIORITY TO 7
  (3)  022060  010400                               MOV     R4,R0
  (3)  022062  104441                               TRAP    C$SPRI
```

```
6169  022064  013705  002700                  MOV     STAT1,R5        ;GET BR LEVEL OF M8200,4,7
6170  022070  006205                          ASR     R5              ;SHIFT R5 4 TIMES
6171  022072  006205                          ASR     R5              ;TO GET PROPER LEVEL
6172  022074  006205                          ASR     R5
6173  022076  006205                          ASR     R5
6174  022100  042705  177437                  BIC     #177437,R5      ;CLEAR UNWANTED BITS
6175  022104  010537  002636                  MOV     R5,$GDDAT
6176  022110  004537  003552                  JSR     R5,SETVEC       ;SET UP VECTORS
6177  022114  022160                          2$                      ;A VECTOR
6178  022116  022160                          2$                      ;B VECTOR
6179  022120  000340  000340                  .WORD   340,340         ;PRIORITY 7
6180  022124  012761  000200  000004  4$:     MOV     #200,4(R1)      ;LOAD PORT4
6181  022132                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1) 022132  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6182  022136  121111                          121111                  ;SET BR REQUEST
6183  022140                          5$:     SETPRI  R4              ;PUT LEVEL IN R2 IN PS
  (3) 022140  010400                          MOV     R4,R0
  (3) 022142  104441                          TRAP    C$SPRI
6184  022144  000240                          NOP
6185  022146  020504                          CMP     R5,R4           ;IS PRESENT PS LEVEL = TO M8200,4,7 LEVEL
6186  022150  001420                          BEQ     1$              ;BR IF YES
6187  022152  162704  000040                  SUB     #40,R4          ;NO GET NEXT LOWER LEVEL IN R2
6188  022156  000770                          BR      5$              ;AND CONTINUE WITH TEST
6189  022160                          2$:     BRESET
  (3) 022160  104433                          TRAP    C$RESET
6190  022162                                  ERROR   33              ;ERROR UNEXPECTED INTERRUPT
  (5) 022200  104455                          TRAP    C$ERDF
  (6) 022202  000041                          .WORD   33
  (6) 022204  005400                          .WORD   EM33
  (6) 022206  010220                          .WORD   ERR33
6191  022210  000002                          RTI
6192  022212                          1$:     MSTCLR
  (1) 022212  004537  003156                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
6193  022216                                  ENDTST
  (3) 022216                          L10114:
  (3) 022216  104401                          TRAP    C$ETST
6194
6195  022220                                  BADHEAD
  (2)                                       ;********************* TEST 35 *********************
6196                                        ;*PRIORITY INTERRUPT TESTS
6197                                        ;*SET PS TO ALL BR LEVELS LESS THAN THE M8200,4,7 LEVEL
6198                                        ;*VERIFY THAT M8200,4,7 WILL INTERRUPT
6199  022220                                BADHEAD
  (2)                                       ;********************* TEST 35 *********************
6200
6201  022220                          BGNTST
  (3) 022220                          T35::
6202  022220                                  MYINT
  (1) 022220  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6203  022224                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1) 022224  004537  003156                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
6204  022230  012704  000340                  MOV     #340,R4         ;PUT LEVEL 7 IN R2
6205  022234                                  SETPRI  R4              ;SET PRIORITY TO 7
  (3) 022234  010400                          MOV     R4,R0
  (3) 022236  104441                          TRAP    C$SPRI
6206  022240  013705  002700                  MOV     STAT1,R5        ;GET BR LEVL  OF M8200,4,7
```

```
6207  022244  006205                        ASR     R5              ;SHIFT R5 4 TIMES
6208  022246  006205                        ASR     R5              ;TO GET PROPER LEVEL
6209  022250  006205                        ASR     R5
6210  022252  006205                        ASR     R5
6211  022254  042705  177437              BIC     #177437,R5      ;CLEAR UNWANTED BITS
6212  022260  010502                        MOV     R5,R2           ;PUT M8200,4,7 LEVEL IN R2
6213  022262  162702  000040              SUB     #40,R2          ;GET NEXT LOWER LEVEL IN R2
6214  022266  004537  003552              JSR     R5,SETVEC       ;SET UP VECTORS
6215  022272  022354                        2$                      ;A VECTOR
6216  022274  022362                        3$                      ;B VECTOR
6217  022276  000340  000340              .WORD   340,340         ;PRIORITY 7
6218  022302  012761  000200  000004  4$: MOV     #200,4(R1)      ;LOAD PORT4
6219  022310                              ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BRN
 (1)  022310  004537  003244              JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6220  022314  121111                        121111                  ;SET BR REQUEST
6221  022316                        5$:   SETPRI  R2              ;PUT LEVEL IN R2 IN PS
 (3)  022316  010200                        MOV     R2,R0
 (3)  022320  10444'                        TRAP    C$SPRI
6222  022322  000240                        NOP
6223  022324                              ERROR   31              ;ERROR, NO INTERRUPT
 (5)  0223'2  104455                        TRAP    C$ERDF
 (6)  022344  000037                        .WORD   31
 (6)  022346  005312                        .WORD   EM31
 (6)  022350  010144                        .WORD   ERR31
6224  022352  000421                  6$:   BR      1$
6225  022354  012716  022352        2$:   MOV     #6$,(SP)        ;SET UP FOR RTI
6226  022360  000002                        RTI
6227  022362                        3$:   ERROR   32              ;ERROR, WRONG VECTOR
 (5)  022400  104455                        TRAP    C$ERDF
 (6)  022402  000040                        .WORD   32
 (6)  022404  005341                        .WORD   EM32
 (6)  022406  010172                        .WORD   ERR32
6228  022410  012716  022416              MOV     #1$,(SP)        ;SET UP FOR RTI
6229  022414  000002                        RTI
6230  022416                        1$:   MSTCLR
 (1)  022416  004537  003156              JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
6231  022422                              ENDTST
 (3)  0224'?                        L10115:
 (3)  022422  104401                        TRAP    C$ETST
6232
6233  022424                              BADHEAD
 (2)                                        ;********** TEST 36 **********
6234                                        ;*NPR TEST
6235                                        ;*TEST OF DATO, 1 WORD FROM UPROC TO 11 MEMORY
6236  022424                              BADHEAD
 (2)                                        ;********** TEST 36 **********
6237
6238  022424                        B.NTST
 (3)  022424                        T36::
6239  022424                              BRESET                  ;BUS RESET
 (3)  022424  104433                        TRAP    C$RESET
6240
6241  022426                              MYINT
 (1)  022426  013701  002716              MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6242  022432  005011                        CLR     (R1)            ;CLEAR RUN
6243  022434  005061  000004              CLR     4(R1)           ;CLR PORT4
```

```
6244  022440  004537  003574          JSR     R5,NPRSET       ;SET UP IBUS REG 0-7
6245  022444  000000                  0                       ;IN DATA
6246  022446  177777                  -1                      ;OUT DATA
6247  022450  022564                  3$                      ;IN BA
6248  022452  022562                  2$                      ;OUT BA
6249  022454  005037  022562          CLR     2$              ;CLEAR 2$
6250  022460  005061  000004          CLR     4(R1)           ;CLEAR PORT 4
6251  022464                          ROMCLK                  ;NOW MOVE TO IBUS*<11>
 (1)  022464  004537  003244          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6252  022470  121111                  121111
6253  022472  012761  000021  000004  MOV     #21,4(R1)       ;WRITE PORT4
6254  022500                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  022500  004537  003244          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6255  022504  121110                  121110                  ;SET NPR BITS IN IBUS* REG 10
6256  022506  000240                  NOP
6257  022510  012737  177777  002636  MOV     #-1,$GDDAT              ;PUT 'EXPECTED'' IN $GDDAT
6258  022516  013704  022562          MOV     2$,R4           ;PUT 'FOUND'' IN R4
6259  022522  023704  002636          CMP     $GDDAT,R4               ;DATA CORRECT?
6260  022526  001413                  BEQ     4$              ;BR IF YES
6261  022530                          ERROR   11,YES          ;ERROR NPR FAILED
 (5)  022542  104455                  TRAP    C$ERDF
 (6)  022544  000013                  .WORD   11
 (6)  022546  004654                  .WORD   EM11
 (6)  022550  006660                  .WORD   ERR11
6262  022552                          ESCAPE  TST
 (3)  022552  104410                  TRAP    C$ESCAPE
 (3)  022554  000012                  .WORD   L10116-.
6263  022556                  4$:     EXIT    TST
 (3)  022556  104432                  TRAP    C$EXIT
 (3)  022560  000006                  .WORD   L10116-.
6264  022562  000000                  2$:     0               ;OUT BA
6265  022564  000000                  3$:     0               ;IN BA
6266  022566                          ENDTST
 (3)  022566                          L10116:
 (3)  022566  104401                          TRAP    C$ETST
6267
6268  022570                          BADHEAD
 (2)                                  ;********** TEST 37 **********
6269                                  ;*NPR TEST
6270                                  ;*TEST OF DATI, 1 WORD FROM 11 MEMORY TO UPROC
6271  022570                          BADHEAD
 (2)                                  ;********** TEST 37 **********
6272
6273  022570                  BGNTST
 (3)  022570                  T37::
6274  022570                          MYINT
 (1)  022570  013701  002716          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6275  022574                          MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  022574  004537  003156          JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
6276  022600  005061  000004          CLR     4(R1)           ;CLR PORT4
6277  022604  004537  003574          JSR     R5,NPRSET       ;SET UP IBUS REG 0-7
6278  022610  000000                  0                       ;IN DATA
6279  022612  177777                  -1                      ;OUT DATA
6280  022614  022734                  3$                      ;IN BA
6281  022616  022732                  2$                      ;OUT BA
6282  022620  012737  177777  022734  MOV     #-1,3$          ;PUT DATA IN 3$
```

```
 6283  022626  012761  000001  000004          MOV     #1,4(R1)        ;WRITE PORT4
 6284  022634                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  022634  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6285  022640  121110                          121110                  ;SET NPR BITS IN IBUS* REG 11
 6286  022642  000240                          NOP
 6287  022644  012737  177777  002636          MOV     #-1,$GDDAT              ;PUT 'EXPECTED'' IN $GDDAT
 6288  022652                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  022652  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6289  022656  021004                          021004                  ;MOVE IN DATA LOW BYTE TO PORT4
 6290  022660                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  022660  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6291  022664  021025                          021025                  ;MOVE IN DATA HIGH BYTE TO PORT5
 6292  022666  016104  000004                  MOV     4(R1),R4        ;PUT 'FOUND'' IN R4
 6293  022672  023704  002636                  CMP     $GDDAT,R4               ;DATA CORRECT?
 6294  022676  001413                          BEQ     4$              ;BR IF YES
 6295  022700                                  ERROR   11,YES          ;ERROR NPR FAILED
  (5)  022712  104455                          TRAP    C$ERDF
  (6)  022714  000013                          .WORD   11
  (6)  022716  004654                          .WORD   EM11
  (6)  022720  006660                          .WORD   ERR11
 6296  022722                                  ESCAPE  TST
  (3)  022722  104410                          TRAP    C$ESCAPE
  (3)  022724  000012                          .WORD   L10117-.
 6297  022726                          4$:     EXIT    TST
  (3)  022726  104432                          TRAP    C$EXIT
  (3)  022730  000006                          .WORD   L10117-.
 6298  022732  000000                          2$:     0                       ;OUT BA
 6299  022734  000000                          3$:     0                       ;IN BA
 6300  022736                                  ENDTST
  (3)  022736                                  L10117:
  (3)  022736  104401                          TRAP    C$ETST
 6301
 6302  022740                                  BADHEAD
  (2)                                         ;********************* TEST 38 *********************
 6303                                         ;*NPR TEST
 6304                                         ;*TEST OF DATOB, 1 BYTE FROM UPROC TO 11 MEMORY
 6305  022740                                  BADHEAD
  (2)                                         ;********************* TEST 38 *********************
 6306
 6307  022740                          BGNTST
  (3)  022740                          T38::
 6308  022740                                  MYINT
  (1)  022740  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6309  022744                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  022744  004537  003156                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 6310  022750  005061  000004                  CLR     4(R1)           ;CLR PORT4
 6311  022754  004537  003574                  JSR     R5,NPRSET       ;SET UP IBUS REG 0-7
 6312  022760  000000                          0                       ;IN DATA
 6313  022762  177777                          -1                      ;OUT DATA
 6314  022764  023100                          3$                      ;IN BA
 6315  022766  023077                          2$+1                    ;OUT BA
 6316  022770  005037  023076                  CLR     2$              ;CLEAR 2$
 6317  022774  005061  000004                  CLR     4(R1)           ;CLEAR PORT 4
 6318  023000                                  ROMCLK                  ;NOW MOVE IT TO IBUS*<11>
  (1)  023000  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6319  023004  121111                          121111
```

```
 6320  023006  012761  000221  000004          MOV     #221,4(R1)      ;WRITE PORT4
 6321  023014                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  023014  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6322  023020  121110                          121110                  ;SET NPR BITS IN IBUS* REG 11
 6323  023022  000240                          NOP
 6324  023024  012737  177400  002636          MOV     #177400,$GDDAT  ;PUT "EXPECTED" IN $GDDAT
 6325  023032  013704  023076                  MOV     2$,R4           ;PUT "FOUND" IN R4
 6326  023036  023704  002636                  CMP     $GDDAT,R4               ;DATA CORRECT?
 6327  023042  001413                          BEQ     4$              ;BR IF YES
 6328  023044                                  ERROR   11,YES          ;ERROR NPR FAILED
  (5)  023056  104455                          TRAP    C$ERDF
  (6)  023060  000013                          .WORD   11
  (6)  023062  004654                          .WORD   EM11
  (6)  023064  006660                          .WORD   ERR11
 6329  023066                                  ESCAPE  TST
  (3)  023066  104410                          TRAP    C$ESCAPE
  (3)  023070  000012                          .WORD   L10120-.
 6330  023072                          4$:     EXIT    TST
  (3)  023072  104432                          TRAP    C$EXIT
  (3)  023074  000006                          .WORD   L10120-.
 6331  023076  000000                  2$:     0                       ;OUT BA
 6332  023100  000000                  3$:     0                       ;IN BA
 6333  023102                          ENDTST
  (3)  023102                          L10120:
  (3)  023102  104401                          TRAP    C$ETST
 6334
 6335  023104                                  BADHEAD
  (2)                                         ;****************** TEST 39 ******************
 6336                                         ;*TEST OF EA BITS 16 AND 17
 6337                                         ;*DO A DATO TO AN ADDRESS USING OUT BA BITS 16 AND 17
 6338                                         ;*VERIFY CORRECT RESULTS
 6339  023104                                  BADHEAD
  (2)                                         ;****************** TEST 39 ******************
 6340
 6341  023104                          BGNTST
  (3)  023104                          T39::
 6342  023104                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  023104  004537  003156                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 6343  023110                                  MYINT
  (1)  023110  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6344  023114  013737  002726  023142          MOV     KMP06,1$        ;USE SEL4 FOR ADDRESS
 6345  023122  013737  002726  023140          MOV     KMP06,2$        ;USE SEL4 FOR ADDRESS
 6346  023130  004537  003574                  JSR     R5,NPRSET       ;LOAD BA AND DATA
 6347  023134  000000                          0                       ;IN DATA
 6348  023136  125252                          125252                  ;OUT DATA
 6349  023140  000000                  2$:     0                       ;IN BA
 6350  023142  000000                  1$:     0                       ;OUT BA
 6351  023144  012761  000014  000004          MOV     #14,4(R1)       ;LOAD SEL 4 WITH OUT BA16 AND 17
 6352  023152                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  023152  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6353  023156  121111                          121111                  ;SET OUTBA 16 AND 17
 6354  023160  012761  000021  000004          MOV     #21,4(R1)       ;LOAD SEL4
 6355  023166  012711  003000                  MOV     #BIT9.BIT10,(R1)
 6356  023172  012761  121110  000006          MOV     #121110,6(R1)   ;PUT INSTRUCTION IN SEL6
 6357  023200  052711  000400                  BIS     #BIT8,(R1)      ;CLOCK IT!
 6358  023204  000240                          NOP                     ;WAIT FOR NPR
```

```
 6359  023206  012737  121110  002636          MOV     #121110,$GDDAT  ;PUT 'EXPECTED' IN $GDDAT
 6360  023214  000240                          NOP
 6361  023216  000240                          NOP
 6362                                                                  ;OK,LISTEN UP.EXPLAINATION TIME.
 6363                                                                  ;
 6364                                                                  ;ON THE NPR OUT,THE DATA ENDED UP
 6365                                                                  ;IN THE IBUS(NOT IBUS*) SENCE SEL A
 6366                                                                  ;WAS ONLY SELECTED IN THE NPR CYCLE.
 6367                                                                  ;THAT IS,WE DIDN'T REALLY DO AN NPR TO
 6368                                                                  ;PORT 6,THE NPR OUT REALLY ENDED UP IN
 6369                                                                  ;OUT DATA LOW,AND OUT DATA HIGH
 6370                                                                  ;(IBUS <2> AND IBUS <3>).
 6371
 6372                                                                  ;WHAT WE'RE DOING NEXT IS READING IBUS 2&3
 6373                                                                  ;TO SEE IF THE DATA GOT XFERRED CORRECTLY.
 6374  023220                                  ROMCLK
  (1)  023220  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6375  023224  021044                          021044                  ;READ IBUS <2> PUT IN PORT 4
 6376  023226                                  ROMCLK
  (1)  023226  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6377  023232  021065                          021065                  ;READ IBUS <3> PUT IN PORT 5
 6378  023234  016104  000004                  MOV     4(R1),R4        ;PUT 'FOUND' IN R4
 6379  023240  023704  002636                  CMP     $GDDAT,R4               ;CORRECT RESULTS?
 6380  023244  001411                          BEQ     3$              ;BR IF YES
 6381  023246                                  ERROR   11,YES          ;ERROR BA 16 AND 17 FAILED
  (5)  023260  104455                          TRAP    C$ERDF
  (6)  023262  000013                          .WORD   11
  (6)  023264  004654                          .WORD   EM11
  (6)  023266  006660                          .WORD   ERR11
 6382  023270                          3$:
 6383  023270                          ENDTST
  (3)  023270                          L10121:
  (3)  023270  104401                          TRAP    C$ETST
 6384
 6385  023272                                  BADHEAD
  (2)                                         ;******************** TEST 40 ********************
 6386                                         ;*TEST OF EA BITS 16 AND 17
 6387                                         ;*DO A DATI USING IN BA BITS 16 AND 17
 6388                                         ;*VERIFY CORRECT RESULTS
 6389                                         ;*IN ORDER TO DO THIS TEST, WE WILL READ THE DATA FROM THE
 639C                                         ;*CONSOL TTY CSR IF ONE EXSITS
 6391                                         ;*IF NO CONSOL TTY CSR AT ADDRESS 177560, THIS TEST
 6392                                         ;*WILL BE SKIPPED
 6393  023272                                  BADHEAD
  (2)                                         ;******************** TEST 40 ********************
 6394
 6395  023272                          BGNTST
  (3)  023272                          T40::
 6396  023272                                  MYINT
  (1)  023272  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6397  023276                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  023276  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 6398  023302  012737  023464  000004  STOP:  MOV     #TOUTT,4        ;SET UP FOR TRAP IN CASE IF NO
 6399  023310  012737  000340  000006          MOV     #340,6  ;TTY AT ADDRESS 177560
 6400  023316  005737  177560                  TST     177560          ;ADDRESS THE TTY-TRAPS HERE IF NONE.
 6401  023322  012737  177560  023350          MOV     #177560,1$      ;USE SEL4 FOR ADDRESS
```

```
6402  023330  012737  177560  023346          MOV     #177560,2$      ;USE SEL4 FOR ADDRESS
6403  023336  004537  003574                  JSR     R5,NPRSET       ;LOAD BA AND DATA
6404  023342  000000                          0                       ;IN DATA
6405  023344  125252                          125252                  ;OUT DATA
6406  023346  000000                  2$:     0                       ;IN BA
6407  023350  000000                  1$:     0                       ;OUT BA
6408  023352  012761  000015  000004          MOV     #15,4(R1)
6409  023360  012711  003000                  MOV     #BIT9.BIT10,(R1);SET CROMI AND CROMO!.
6410  023364  012761  121110  000006          MOV     #121110,6(R1)   ;PUT INSTR INTO SEL6  NW*
6411  023372  052711  000400                  BIS     #BIT8,(R*)      ;CLOCK IT.
6412  023376  000240                          NOP                     ;WAIT FOR NPR
6413  023400                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  023400  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6414  023404  021004                          021004                  ;MOVE OUT DATA LB TO SEL4
6415  023406                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  023406  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6416  023412  021025                          02*025                  ;MOVE OUT DATA HB TO SEL5
6417  023414  016104  000004                  MOV     4(R1),R4        ;PUT 'FOUND" IN R4
6418  023420  013737  177560  002636          MOV     177560,$GDDAT
6419  023426  042737  000200  002636          BIC     #200,$GDDAT
6420  023434  023704  002636                  CMP     $GDDAT,R4               ;CORRECT RESULTS?
6421  023440  001413                          BEQ     TOUTP           ;BR IF YES
6422  023442                                  ERROR   11,YES          ;ERROR BA 16 AND 17 FAILED
 (5)  023454  104455                          TRAP    C$ERDF
 (6)  023456  000013                          .WORD   11
 (6)  023460  004654                          .WORD   EM11
 (6)  023462  006660                          .WORD   ERR11
6423  023464                          3$:
6424  023464  062706  000004          TOUTT:  ADD     #4,SP           ;UPDATE STACK POINTER
6425  023470  013737  002652  000006  TOUTP:  MOV     SAVE6,6         ;RESTORE TRAP VECTOR
6426  023476  013737  002650  000004          MOV     SAVE4,4
6427  023504                          ENDTST
 (3)  023504                          L10122:
 (3)  023504  104401                          TRAP    C$ETST
6428
6429  023506                                  BADHEAD
 (2)                                          ;****************** TEST 41 ******************
```

```
6431                                                ;*NPR NON-EXISTENT MEMORY TEST
6432                                                ;*DO A DATO TO A NON-EXISTENT ADDRESS
6433                                                ;*VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
6434  023506                                        BADHEAD
 (2)                                                ;****************** TEST 41 ******************
6435
6436  023506                                  BGNTST
 (3)  023506                                  T41::
6437  023506                                        MYINT
 (1)  023506  013701  002716                      MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6438  023512                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  023512  004537  003156                      JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
6439  023516  004537  003574                      JSR     R5,NPRSET       ;LOAD IBUS REGISTERS 0-7
6440  023522  000000                              0                       ;IN DATA
6441  023524  000000                              0                       ;OUT DATA
6442  023526  177320                            177320                    ;IN BA
6443  023530  177320                            177320                    ;IN BA
6444  023532  012761  000014  000004            MOV     #14,4(R1)       ;SET OUT BA BITS 16+17 IN PORT4
```

```
6446  023540                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  023540  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6447  023544  121111                               121111                  ;SET OUTBA 16 AND 17
6448  023546  012761  000021  000004               MOV     #21,4(R1)       ;SET NPR REQUEST BITS IN PORT4
6449  023554                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  023554  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6450  023560  121110                               121110                  ;MOV IBUS* 4 TO IBUS* 10
6451  023562  000240                               NOP
6452  023564                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 (1)  023564  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6453  023570  121225                               121225                  ;MOV IBUS*11 TO IBUS*5
6454  023572  012737  000001  002636               MOV     #1,$GDDAT               ;PUT 'EXPECTED' IN $GDDAT
6455  023600  116104  000005                       MOVB    5(R1),R4        ;PUT 'FOUND' IN R4
6456  023604  042704  177776                       BIC     #177776,R4      ;CLEAR UNWANTED BITS
6457  023610  023704  002636                       CMP     $GDDAT,R4               ;DATA CORRECT?
6458  023614  001411                               BEQ     1$              ;BR IF YES
6459  023616                                        ERROR   13,YES          ;ERROR NON-EXISTENT MEM BIT FAILED TO SET
 (5)  023630  104455                               TRAP    C$ERDF
 (6)  023632  000015                               .WORD   13
 (6)  023634  004707                               .WORD   EM13
 (6)  023636  007014                               .WORD   ERR13
6460  023640                              1$:
6461  023640  152761  000100  000001               BISB    #100,1(R1)      ;SET MASTER CLEAR
6462  023646  142761  000100  000001               BICB    #100,1(R1)      ;CLEAR MASTER
6463  023654                                        ROMCLK                  ;MOV IBUS*11 TO
 (1)  023654  004537  003244                        JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6464  023660  121225                               121225                  ;PORT5
6465  023662  005037  002636                       CLR     $GDDAT          ;EXPECT CLEAR
6466  023666  116104  000005                       MOVB    5(R1),R4        ;GET NPR REG
6467  023672  042704  177776                       BIC     #177776,R4      ;CLEAR JUNK
6468  023676  001411                               BEQ     2$              ;EXIT IF CLEAR
6469  023700                                        ERROR   13,YES          ;NON-EXISTANT MEM
 (5)  023712  104455                               TRAP    C$ERDF
 (6)  023714  000015                               .WORD   13
 (6)  023716  004707                               .WORD   EM13
 (6)  023720  007014                               .WORD   ERR13
6470                                                                      ;BIT FAILED TO CLEAR
6471  023722                              2$:
6472  023722                              ENDTST
 (3)  023722                              L10123:
 (3)  023722  104401                               TRAP    C$ETST
6473
6474  023724                                        BADHEAD
 (2)                                              ;******************** TEST 42 ********************
6475                                              ;*NPR NON-EXISTENT MEMORY TEST
6476                                              ;*DO A DATI FROM A NON-EXISTENT ADDRESS
6477                                              ;*VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
6478  023724                                        BADHEAD
 (2)                                              ;******************** TEST 42 ********************
6479
6480  023724                              BGNTST
 (3)  023724                              T42::
6481  023724                                        MYINT
 (1)  023724  013701  002716                        MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6482  023730                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  023730  004537  003156                        JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
```

```
 6483  023734  004537  003574          JSR     R5,NPRSET       ;LOAD IBUS REGISTERS 0-7
 6484  023740  000000                  0                       ;IN DATA
 6485  023742  000000                  0                       ;OUT DATA
 6486  023744  177320                  177320                  ;IN BA
 6487  023746  177320                  177320                  ;OUT BA
 6488  023750  005061  000004          CLR     4(R1)
 6489  023754                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
   (1) 023754  004537  003244          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6490  023760  121111                  121111                  ;CLEAR NON-EXISTENT BIT
 6491  023762  012761  000015  000004  MOV     #15,4(R1)       ;SET NPR REQUEST BITS IN PORT4
 6492  023770                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
   (1) 023770  004537  003244          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6493  023774  121110                  121110                  ;MOV IBUS* 4 TO IBUS* 10
 6494  023776  000240                  NOP
 6495  024000                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
   (1) 024000  004537  003244          JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 6496  024004  121225                  121225                  ;MOV IBUS*11 TO IBUS*5
 6497  024006  012737  000001  002636  MOV     #1,$GDDAT               ;PUT 'EXPECTED' IN $GDDAT
 6498  024014  116104  000005          MOVB    5(R1),R4        ;PUT 'FOUND' IN R4
 6499  024020  042704  177776          BIC     #177776,R4      ;CLEAR UNWANTED BITS
 6500  024024  023704  002636          CMP     $GDDAT,R4               ;DATA CORRECT?
 6501  024030  001411                  BEQ     1$              ;BR IF YES
 6502  024032                          ERROR   13,YES          ;ERROR NON-EXISTENT MEM BIT FAILED TO SET
   (5) 024044  104455                  TRAP    C$ERDF
   (6) 024046  000015                  .WORD   13
   (6) 024050  004707                  .WORD   EM13
   (6) 024052  007014                  .WORD   ERR13
 6503  024054                  1$:
 6504  024054                  ENDTST
   (3) 024054                  L10124:
   (3) 024054  104401                  TRAP    C$ETST
 6505
 6506  024056                          BADHEAD
   (2)                                   ;********************* TEST 43 *********************
 6507                                    ;*NPR TEST
 6508                                    ;*USING DATO, NPR A BINARY COUNT (0-377)
 6509                                    ;*FROM MICRO-PROCESSOR TO ALL AVAILABLE MEMORY
 6510  024056                          BADHEAD
   (2)                                   ;********************* TEST 43 *********************
 6511
 6512  024056                  BGNTST
   (3) 024056                  T43::
 6513  024056                          MYINT
   (1) 024056  013701  002716          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6514  024062                          MSTCLR                  ;MASTER CLEAR M8200,4,7
   (1) 024062  004537  003156          JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 6515  024066  005037  024270          CLR     5$              ;START FLAG AT 0
 6516  024072  005005                  CLR     R5              ;DATA
 6517  024074  012702  035404          MOV     #CORMAX,R2      ;ADDRESS
 6518  024100                  1$:
 6519  024100  010537  024130          MOV     R5,2$           ;LOAD DATA
 6520  024104  010237  024134          MOV     R2,4$           ;LOAD BA
 6521  024110  032702  000001          BIT     #BIT0,R2        ;IS BA ODD?
 6522  024114  001402                  BEQ     .+6             ;BR IF NO
 6523  024116  000337  024130          SWAB    2$              ;IF ODD PUT DATA IN HI-BYTE
 6524  024122  004537  003574          JSR     R5,NPRSET       ;LOAD NPR REGISTERS
```

```
024126  000000                       0                          ;IN DATA
024130  000000                  2$:  0                          ;OUT DATA
024132  000000                       0                          ;IN BA
024134  000000                  4$:  0                          ;OUT BA
024136  105012                       CLRB    (R2)               ;CLEAR MEMORY LOCATION
024140  012761  000221  000004       MOV     #221,4(R1)         ;LOAD PORT4
024146                               ROMCLK                     ;NEXT WORD IS INSTRUCTION,  BBN
024146  004537  003244               JSR     R5,.ROMCLK                 ;CLOCK INSTRUCTION
024152  121110                       121110                     ;DO THE NPR
024154  000240                       NOP
024156  010537  002636               MOV     R5,$GDDAT                  ;PUT 'EXPECTED'' IN $GDDAT
024162  111204                       MOVB    (R2),R4 ;PUT 'FOUND'' IN R4
024164  123704  002636               CMPB    $GDDAT,R4                  ;IS DATA CORRECT?
024170  001411                       BEQ     3$                 ;BR IF YES
024172                               ERROR   11,YES             ;ERROR, DATA INCORRECT
024204  104455                       TRAP    C$ERDF
024206  000013                       .WORD   11
024210  004654                       .WORD   EM11
024212  006660                       .WORD   ERR11
024214                          3$:  ESCAPE  TST
024214  104410                       TRAP    C$ESCAPE
024216  000054                       .WORD   L10125-.
024220  005205                       INC     R5                 ;NEXT CHARACTER
024222  042705  177400               BIC     #177400,R5         ;USE ONLY LOW BYTE
024226  005737  024270               TST     5$                 ;HAS MAX MEMORY BEEN REACHED YET?
024232  001402                       BEQ     6$                 ;BR IF NO
024234  005705                       TST     R5                 ;DONE PATTERN?
024236  001412                       BEQ     7$                 ;BR IF YES
024240  005202                  6$:  INC     R2                 ;INC BA
024242  023702  002604               CMP     MEMLIM,R2          ;REACHED MEMORY LIMIT YET?
024246  001314                       BNE     1$                 ;BR IF NOT
024250  012702  035404               MOV     #CORMAX,R2         ;RESTART BA AT FIRST ADDRESS
024254  012737  177777  024270       MOV     #-1,5$             ;SET FLAG TO END TEST AT END OF DATA PATTERN
024262  000706                       BR 1$                      ;CONTINUE
024264                          7$:
024264                               EXIT    TST
024264  104432                       TRAP    C$EXIT
024266  000004                       .WORD   L10125-.
024270  000000                  5$:  0                          ;THIS LOCATION IS A FLAG, IT STARTS AT 0,
                                                                ;AND IS SET TO -1 WHEN LAST MEMORY ADDRESS
                                                                ;IS USED, TEST IS THEN ENDED WHEN PATTERN IS FINISHED
024272                          ENDTST
024272                          L10125:
024272  104401                       TRAP    C$ETST
                                ;$MEM1
                                ;$MEM0
                                ;$MEM2  1K
                                ;$MEM3  1K

024274                               BADHEAD
                                     ;******************* TEST 44 ********************
                                     ;*ALU C BIT TEST
                                     ;*TEST THAT AN ADD OF 377 AND 377 WILL SET THE C BIT
024274                               BADHEAD
                                     ;******************* TEST 44 ********************
```

```
 6568  024274                                  BGNTST
  (3)  024274                                  T44::
 6569  024274                                          MYINT
  (1)  024274  013701  002716                         MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 6570  024300                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  024300  004537  003156                         JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 6571  024304  004737  003640                         JSR     PC,MEMLD        ;LOAD MAINMEM DATA
 6572  024310  024424                              TDATA                   ;POINTER TO DATA
 6573  024312  004737  004012                         JSR     PC,SPLD         ;LOAD SP DATA
 6574  024316  024424                              TDATA                   ;POINTER TO DATA
 6575  024320                                          BGNSEG
  (3)  024320  104404                                TRAP    C$BSEG
 6576  024322                                  1$:
 6577  024322                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  024322  004537  003244                         JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6578  024326  010000                              010000                  ;MAR 0
 6579  024330                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  024330  004537  003244                         JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6580  024334  054400                              054400!<0*20>           ;ADD 377 AND 377, TO SET C BIT
 6581  024336                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  024336  004537  003244                         JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6582  024342  040421                              040401.<1*20>           ;ADD 0 AND 0 AND THE C BIT
 6583  024344                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
  (1)  024344  004537  003244                         JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 6584  024350  061224                              61224                   ;PUT RESULTS IN PORT4
 6585  024352  012737  000001  002636                 MOV     #1,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
 6586  024360  016104  000004                         MOV     4(R1),R4        ;PUT "FOUND" IN R4
 6587  024364  123704  002636                         CMPB    $GDDAT,R4               ;DATA CORRECT?
 6588  024370  001411                                 BEQ     2$              ;BR IF YES
 6589  024372                                          ERROR   34,YES          ;ERROR C BIT NOT SET
  (5)  024404  104455                                TRAP    C$ERDF
  (6)  024406  000042                                .WORD   34
  (6)  024410  005435                                .WORD   EM34
  (6)  024412  010276                                .WORD   ERR34
 6590  024414                                  2$:     ESCAPE  SEG
  (3)  024414  104410                                TRAP    C$ESCAPE
  (3)  024416  000002                                .WORD   10000$-.
 6591  024420                                          ENDSEG
  (3)  024420                                  10000$:
  (3)  024420  104405                                TRAP    C$ESEG
 6592  024422                                  ENDTST
  (3)  024422                                  L10126:
  (3)  024422  104401                                TRAP    C$ETST
 6593  024424     377     000     000  TDATA:  .BYTE   -1,0,0,0,0,0,0,0
       024427     000     000     000
       024432     000     000
 6594
 6595                                         .EVEN
 6596
 6597  024434                                          BADHEAD
  (2)                                             ;********************* TEST 45 *********************
 6598                                             ;*ALU TEST
 6599                                             ;*TEST OF ALU FUNCTION SEL B WITH C BIT CLEARED
 6600                                             ;*ALU FUNCTION (B)    CODE=11
 6601                                             ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 6602                                             ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
```

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

                                BGNTST
                                T45::
                                        MYINT
 (1)  024434  013701  002716            MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  024440  004537  003156            JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
                                        CLR     R5              ;MEM + SP ADDRESS
                                        MOV     #5$,R2          ;POINTER TO CORRECT DATA
                                        JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
                                        MEMDAT                  ;POINTER TO DATA
                                        JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
                                        SPDAT                   ;POINTER TO DATA
                                        BGNSEG
 (3)  024466  104404                    TRAP    C$BSEG
                                1$:     JSR     PC,CLRC         ;CLEAR C BIT!
                                        BIC     #^C7,2$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
                                        BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  024506  004537  003244            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
                                2$:     010000                  ;LOAD MAR
                                        BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
                                        BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  024526  004537  003244            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
                                3$:     040400!<11*20>          ;BR   SEL B
                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  024534  004537  003244            JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
                                        61224                   ;MOVE BR TO PORT4
                                        MOVB    (R2),$GDDAT             ;PUT 'EXPECTED' IN $GDDAT
                                        MOVB    4(R1),R4        ;PUT 'FOUND' IN R4
                                        CMPB    $GDDAT,R4               ;DATA CORRECT?
                                        BEQ     4$              ;BR IF YES
                                        ERROR   15,YES          ;ALU ERROR
 (5)  024572  104455                    TRAP    C$ERDF
 (6)  024574  000017                    .WORD   15
 (6)  024576  004754                    .WORD   EM15
 (6)  024600  007120                    .WORD   ERR15
                                4$:     ESCAPE  SEG
 (3)  024602  104410                    TRAP    C$ESCAPE
 (3)  024604  000014                    .WORD   10000$-.
                                        INC     R2              ;NEXT DATA
                                        INC     R5              ;NEXT ADDRESS
                                        CMP     #10,R5          ;DONE YET?
                                        BNE     1$              ;BR IF NO
                                        ENDSEG
                                10000$:
 (3)  024620  104405                    TRAP    C$ESEG
                                        EXIT    TST
 (3)  024622  104432                    TRAP    C$EXIT
 (3)  024624  000012                    .WORD   L10127-.
                                5$:     .BYTE   0,-1,0,-1,125,252,125,252
      024626    000   377   000
      024631    377   125   252
      024634    125   252
```

```
6639
6640                                    .EVEN
6641  024636                            ENDTST
 (3)  024636                            L10127:
 (3)  024636  104401                            TRAP    C$ETST
6642
6643  024640                                    BADHEAD
 (2)                                            ;****************** TEST 46 ******************
6644                                            ;*ALU TEST
6645                                            ;*TEST OF ALU FUNCTION SEL A WITH C BIT CLEARED
6646                                            ;*ALU FUNCTION (A)    CODE=10
6647                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6648                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
6649  024640                                    BADHEAD
 (2)                                            ;****************** TEST 46 ******************
6650
6651  024640                            BGNTST
 (3)  024640                            T46::
6652  024640                                    MYINT
 (1)  024640  013701  002716                    MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6653  024644                                    MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  024644  004537  003156                    JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
6654  024650  005005                            CLR     R5              ;MEM + SP ADDRESS
6655  024652  012702  025032                    MOV     #5$,R2          ;POINTER TO CORRECT DATA
6656  024656  004737  003640                    JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
6657  024662  002654                            MEMDAT                  ;POINTER TO DATA
6658  024664  004737  004012                    JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
6659  024670  002664                            SPDAT                   ;POINTER TO DATA
6660  024672                                    BGNSEG
 (3)  024672  104404                            TRAP    C$BSEG
6661  024674  004737  004060            1$:     JSR     PC,CLRC         ;CLEAR C BIT!
6662  024700  042737  000017  024716            BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
6663  024706  050537  024716                    BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
6664  024712                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  024712  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6665  024716  010000                    2$:     010000                  ;LOAD MAR
6666  024720  042737  000017  024736            BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
6667  024726  050537  024736                    BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
6668  024732                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  024732  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6669  024736  040600                    3$:     040400!<10*20>          ;BR   SEL A
6670  024740                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  024740  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6671  024744  061224                            61224                   ;MOVE FR TO PORT4
6672  024746  111237  002636                    MOVB    (R2),$GDDAT             ;PUT 'EXPECTED'' IN $GDDAT
6673  024752  116104  000004                    MOVB    4(R1),R4        ;PUT 'FOUND'' IN R4
6674  024756  123704  002636                    CMPB    $GDDAT,R4               ;DATA CORRECT?
6675  024762  001411                            BEQ     4$              ;BR IF YES
6676  024764                                    ERROR   15,YES          ;ALU ERROR
 (5)  024776  104455                            TRAP    C$ERDF
 (6)  025000  000017                            .WORD   15
 (6)  025002  004754                            .WORD   EM15
 (6)  025004  007120                            .WORD   ERR15
6677  025006                            4$:     ESCAPE  SEG
 (3)  025006  104410                            TRAP    C$ESCAPE
 (3)  025010  000014                            .WORD   10000$-.
```

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


                                        .EVEN
                                        ENDTST
                                        L10130:
                                                TRAP    C$ETST

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

                                        BGNTST
                                        T47::
                                                MYINT
 (1)  025044  013701  002716                    MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
                                                MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  025050  004537  003156                    JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
                                                CLR     R5              ;MEM + SP ADDRESS
                                                MOV     #5$,R2          ;POINTER TO CORRECT DATA
                                                JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
                                                MEMDAT                  ;POINTER TO DATA
                                                JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
                                                SPDAT                   ;POINTER TO DATA
                                                BGNSEG
 (3)  025076  104404                            TRAP    C$BSEG
                                        1$:     JSR     PC,CLRC         ;CLEAR C BIT!
                                                BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
                                                BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
                                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  025116  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
                                        2$:     010000                  ;LOAD MAR
                                                BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
                                                BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
                                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  025136  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
                                        3$:     040400.<12*20>          ;BR   A OR NOTB
                                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  025144  004537  003244                    JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
```

```
6717  025150  061224                          61224                   ;MOVE BR TO PORT4
6718  025152  111237  002636                  MOVB    (R2),$GDDAT             ;PUT 'EXPECTED' IN $GDDAT
6719  025156  116104  000004                  MOVB    4(R1),R4        ;PUT 'FOUND' IN R4
6720  025162  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
6721  025166  001411                          BEQ     4$              ;BR IF YES
6722  025170                                  ERROR   15,YES          ;ALU ERROR
 (5)  025202  104455                          TRAP    C$ERDF
 (6)  025204  000017                          .WORD   15
 (6)  025206  004754                          .WORD   EM15
 (6)  025210  007120                          .WORD   ERR15
6723  025212                          4$:     ESCAPE  SEG
 (3)  025212  104410                          TRAP    C$ESCAPE
 (3)  025214  000014                          .WORD   10000$-.
6724  025216  005202                          INC     R2              ;NEXT DATA
6725  025220  005205                          INC     R5              ;NEXT DATA
6726  025222  027705  000010                  CMP     #10,R5          ;DONE YET?
6727  025226  001324                          BNE     1$              ;BR IF NO
6728  025230                                  ENDSEG
 (3)  025230                          10000$:
 (3)  025230  104405                          TRAP    C$ESEG
6729  025232                                  EXIT    TST
 (3)  025232  104432                          TRAP    C$EXIT
 (3)  025234  000012                          .WORD   L10131-.
6730  025236     377      000      377  5$:     .BYTE   -1,0,-1,-1,-1,125,252,-1
      025241     377      377      125
      025244     252      377
6731
6732                                    .EVEN
6733  025246                            ENDTST
 (3)  025246                            L10131:
 (3)  025246  104401                          TRAP    C$ETST
6734
6735  025250                                  BADHEAD
 (2)                                    ;********************* TEST 48 ******* ************
6736                                    ;*ALU TEST
6737                                    ;*TEST OF ALU FUNCTION A AND B WITH C BIT CLEARED
6738                                    ;*ALU FUNCTION (A AND B)     CODE 13
6739                                    ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6740                                    ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
6741  025250                            BADHEAD
 (2)                                    ;********************* TEST 48 **********************
6742
6743  025250                          BGNTST
 (3)  025250                          T48::
6744  025250                                  MYINT
 (1)  025250  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6745  025254                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  025254  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
6746  025260  005005                          CLR     R5              ;MEM + SP ADDRESS
6747  025262  012702  025442                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
6748  025266  004737  003640                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
6749  025272  002654                          MEMDAT                  ;POINTER TO DATA
6750  025274  004737  004012                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
6751  025300  002664                          SPDAT                   ;POINTER TO DATA
6752  025302                                  BGNSEG
 (3)  025302  104404                          TRAP    C$BSEG
```

```
6753  025304  004737  004060          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
6754  025310  042737  000017  025326          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
6755  025316  050537  025326                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
6756  025322                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1) 025322  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6757  025326  010000                  2$:     010000                  ;LOAD MAR
6758  025330  042737  000017  025346          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
6759  025336  050537  025346                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
6760  025342                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1) 025342  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6761  025346  040660                  3$:     040400!<13*20>          ;BR   A AND B
6762  025350                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1) 025350  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6763  025354  061224                          61224                   ;MOVE BR TO PORT4
6764  025356  111237  002636                  MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
6765  025362  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
6766  025366  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
6767  025372  001411                          BEQ     4$              ;BR IF YES
6768  025374                                  ERROR   15,YES          ;ALU ERROR
  (5) 025406  104455                          TRAP    C$ERDF
  (6) 025410  000017                          .WORD   15
  (6) 025412  004754                          .WORD   EM15
  (6) 025414  007120                          .WORD   ERR15
6769  025416                          4$:     ESCAPE  SEG
  (3) 025416  104410                          TRAP    C$ESCAPE
  (3) 025420  000014                          .WORD   10000$-.
6770  025422  005202                          INC     R2              ;NEXT DATA
6771  025424  005205                          INC     R5              ;NEXT DATA
6772  025426  022705  000010                  CMP     #10,R5          ;DONE YET?
6773  025432  001324                          BNE     1$              ;BR IF NO
6774  025434                                  ENDSEG
  (3) 025434                          10000$:
  (3) 025434  104405                          TRAP    C$ESEG
6775  025436                                  EXIT    TST
  (3) 025436  104432                          TRAP    C$EXIT
  (3) 025440  000012                          .WORD   L10132-.
6776  025442     000      000      000  5$:     .BYTE   0,0,0,-1,125,0,0,252
      025445     377      125      000
      025450     000      252
6777
6778                                  .EVEN
6779  025452                          ENDTST
  (3) 025452                          L10132:
  (3) 025452  104401                          TRAP    C$ETST
6780
6781  025454                                  BADHEAD
  (2)                                     ;****************** TEST 49 ******************
6782                                      ;*ALU TEST
6783                                      ;*TEST OF ALU FUNCTION A OR B WITH C BIT CLEARED
6784                                      ;*ALU FUNCTION (A OR B)     CODE=14
6785                                      ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6786                                      ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
6787  025454                              BADHEAD
  (2)                                     ;****************** TEST 49 ******************
6788
6789  025454                          BGNTST
```

```
   (3)  025454                           T49::
  6790  025454                                   MYINT
   (1)  025454  013701  002716                   MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
  6791  025460                                   MSTCLR                  ;MASTER CLEAR M8200,4,7
   (1)  025460  004537  003156                   JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
  6792  025464  005005                           CLR     R5              ;MEM + SP ADDRESS
  6793  025466  012702  025646                   MOV     #5$,R2          ;POINTER TO CORRECT DATA
  6794  025472  004737  003640                   JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
  6795  025476  002654                           MEMDAT                  ;POINTER TO DATA
  6796  025500  004737  004012                   JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
  6797  025504  002664                           SPDAT                   ;POINTER TO DATA
  6798  025506                                   BGNSEG
   (3)  025506  104404                           TRAP    C$BSEG
  6799  025510  004737  004060           1$:     JSR     PC,CLRC         ;CLEAR C BIT!
  6800  025514  042737  000017  025532           BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
  6801  025522  050537  025532                   BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
  6802  025526                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
   (1)  025526  004537  003244                   JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
  6803  025532  010000                   2$:     010000                  ;LOAD MAR
  6804  025534  042737  000017  025552           BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
  6805  025542  050537  025552                   BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
  6806  025546                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
   (1)  025546  004537  003244                   JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
  6807  025552  040700                   3$:     040400!<14*20>          ;BR   A OR B
  6808  025554                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
   (1)  025554  004537  003244                   JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
  6809  025560  061224                           61224                   ;MOVE BR TO PORT4
  6810  025562  111237  002636                   MOVB    (R2),$GDDAT             ;PUT ''EXPECTED'' IN $GDDAT
  6811  025566  116104  000004                   MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
  6812  025572  123704  002636                   CMPB    $GDDAT,R4               ;DATA CORRECT?
  6813  025576  001411                           BEQ     4$              ;BR IF YES
  6814  025600                                   ERROR   15,YES          ;ALU ERROR
   (5)  025612  104455                           TRAP    C$ERDF
   (6)  025614  000017                           .WORD   15
   (6)  025616  004754                           .WORD   EM15
   (6)  025620  007120                           .WORD   ERR15
  6815  025622                           4$:     ESCAPE  SEG
   (3)  025622  104410                           TRAP    C$ESCAPE
   (3)  025624  000014                           .WORD   10000$-.
  6816  025626  005202                           INC     R2              ;NEXT DATA
  6817  025630  005205                           INC     R5              ;NEXT DATA
  6818  025632  022705  000010                   CMP     #10,R5          ;DONE YET?
  6819  025636  001324                           BNE     1$              ;BR IF NO
  6820  025640                                   ENDSEG
   (3)  025640                           10000$:
   (3)  025640  104405                           TRAP    C$ESEG
  6821  025642                                   EXIT    TST
   (3)  025642  104432                           TRAP    C$EXIT
   (3)  025644  000012                           .WORD   L10133-.
  6822  025646     000     377     377   5$:     .BYTE   0,-1,-1,-1,125,-1,-1,252
        025651     377     125     377
        025654     377     252
  6823
  6824                                   .EVEN
  6825  025656                           ENDTST
   (3)  025656                           L10133:
```

```
 (3)  025656  104401                          TRAP    C$ETST
6826
6827  025660                                 BADHEAD
 (2)                                         ;****************** TEST 50 ******************
6828                                         ;*ALU TEST
6829                                         ;*TEST OF ALU FUNCTION A XOR B WITH C BIT CLEARED
6830                                         ;*ALU FUNCTION (A XOR B)    CODE 15
6831                                         ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6832                                         ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
6833  025660                                 BADHEAD
 (2)                                         ;****************** TEST 50 ******************
6834
6835  025660                          BGNTST
 (3)  025660                          T50::
6836  025660                                 MYINT
 (1)  025660  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6837  025664                                 MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  025664  004537  003156                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
6838  025670  005005                        CLR     R5              ;MEM + SP ADDRESS
6839  025672  012702  026052                MOV     #5$,R2          ;POINTER TO CORRECT DATA
6840  025676  004737  003640                JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
6841  025702  002654                        MEMDAT                  ;POINTER TO DATA
6842  025704  004737  004012                JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
6843  025710  002664                        SPDAT                   ;POINTER TO DATA
6844  025712                                 BGNSEG
 (3)  025712  104404                        TRAP    C$BSEG
6845  025714  004737  004060          1$:   JSR     PC,CLRC         ;CLEAR C BIT!
6846  025720  042737  000017  025736        BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
6847  025726  050537  025736                BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
6848  025732                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
 (1)  025732  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6849  025736  010000                  2$:   010000                  ;LOAD MAR
6850  025740  042737  000017  025756        BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
6851  025746  050537  025756                BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
6852  025752                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  025752  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6853  025756  040720                  3$:   040400!<15*20>          ;BR   A XOR B
6854  025760                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  025760  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
6855  025764  061224                        61224                   ;MOVE BR TO PORT4
6856  025766  111237  002636                MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
6857  025772  116104  000004                MOVB    4(R1),R4        ;PUT "FOUND" IN R4
6858  025776  123704  002636                CMPB    $GDDAT,R4               ;DATA CORRECT?
6859  026002  001411                        BEQ     4$              ;BR IF YES
6860  026004                                 ERROR   15,YES          ;ALU ERROR
 (5)  026016  104455                        TRAP    C$ERDF
 (6)  026020  000017                        .WORD   15
 (6)  026022  004754                        .WORD   EM15
 (6)  026024  007120                        .WORD   ERR15
6861  026026                           4$:   ESCAPE  SEG
 (3)  026026  104410                        TRAP    C$ESCAPE
 (3)  026030  000014                        .WORD   10000$-.
6862  026032  005202                        INC     R2              ;NEXT DATA
6863  026034  005205                        INC     R5              ;NEXT DATA
6864  026036  022705  000010                CMP     #10,R5          ;DONE YET?
6865  026042  001324                        BNE     1$              ;BR IF NO
```

```
      026044                                     ENDSEG
(3)   026044                          10000$:
(3)   026044  104405                             TRAP      C$ESEG
      026046                                     EXIT      TST
(3)   026046  104432                             TRAP      C$EXIT
(3)   026050  000012                             .WORD     L10134-.
      026052     000      377     377 5$:        .BYTE     0,-1,-1,0,0,-1,-1,0
      026055     000      000     377
      026060     377      000


                                      .EVEN
      026062                          ENDTST
(3)   026062                          L10134:
(3)   026062  104401                             TRAP      C$ETST

      026064                                     BADHEAD
(2)                                              ;******************* TEST 51 *******************
                                                 ;*ALU TEST
                                                 ;*TEST OF ALU FUNCTION ADD WITH C BIT CLEARED
                                                 ;*ALU FUNCTION (A PLUS B)     CODE-00
                                                 ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                                 ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
      026064                                     BADHEAD
(2)                                              ;******************* TEST 51 *******************

      026064                          BGNTST
(3)   026064                          T51::
      026064                                     MYINT
(1)   026064  013701  002716                     MOV       KMCSR,R1          ;GET DEVICE ADDRESS.
      026070                                     MSTCLR                      ;MASTER CLEAR M8200,4,7
(1)   026070  004537  003156                     JSR       R5,.MSTCLR                 ;CLEAR M8200,4,7
      026074  005005                             CLR       R5                ;MEM + SP ADDRESS
      026076  012702  026256                     MOV       #5$,R2            ;POINTER TO CORRECT DATA
      026102  004737  003640                     JSR       PC,MEMLD          ;LOAD 8 WORDS OF MAIN MEMORY
      026106  002654                             MEMDAT                      ;POINTER TO DATA
      026110  004737  004012                     JSR       PC,SPLD           ;LOAD 8 WORDS OF SP
      026114  002664                             SPDAT                       ;POINTER TO DATA
      026116                                     BGNSEG
(3)   026116  104404                             TRAP      C$BSEG
      026120  004737  004060          1$:        JSR       PC,CLRC           ;CLEAR C BIT.
      026124  042737  000017  026142             BIC       #17,2$            ;CLEAR ADDRESS FIELD OF INSTRUCTION
      026132  050537  026142                     BIS       R5,2$             ;ADD ADDRESS TO INSTRUCTION
      026136                                     ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)   026136  004537  003244                     JSR       R5,.ROMCLK                 ;CLOCK INSTRUCTION
      026142  010000                  2$:        010000                      ;LOAD MAR
      026144  042737  000017  026162             BIC       #17,3$            ;CLEAR ADDRESS OF INSTRUCTION
      026152  050537  026162                     BIS       R5,3$             ;ADD ADDRESS TO INSTRUCTION
      026156                                     ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)   026156  004537  003244                     JSR       R5,.ROMCLK                 ;CLOCK INSTRUCTION
      026162  040400                  3$:        040400!<00*20>              ;BR   ADD
      026164                                     ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)   026164  004537  003244                     JSR       R5,.ROMCLK                 ;CLOCK INSTRUCTION
      026170  061224                             61224                       ;MOVE BR TO PORT4
      026172  111237  002636                     MOVB      (R2),$GDDAT                ;PUT ''EXPECTED'' IN $GDDAT
      026176  116104  000004                     MOVB      4(R1),R4          ;PUT ''FOUND'' IN R4
      026202  123704  002636                     CMPB      $GDDAT,R4                  ;DATA CORRECT?
```

```
6905  026206  001411                          BEQ     4$              ;BR IF YES
6906  026210                                  ERROR   15,YES          ;ALU ERROR
 (5)  026222  104455                          TRAP    C$ERDF
 (6)  026224  000017                          .WORD   15
 (6)  026226  004754                          .WORD   EM15
 (6)  026230  007120                          .WORD   ERR15
6907  026232                          4$:     ESCAPE  SEG
 (3)  026232  104410                          TRAP    C$ESCAPE
 (3)  026234  000014                          .WORD   10000$-.
6908  026236  005202                          INC     R2              ;NEXT DATA
6909  026240  005205                          INC     R5              ;NEXT DATA
6910  026242  022705  000010                  CMP     #10,R5          ;DONE YET?
6911  026246  001324                          BNE     1$              ;BR IF NO
6912  026250                                  ENDSEG
 (3)  026250                          10000$:
 (3)  026250  104405                          TRAP    C$ESEG
6913  026252                                  EXIT    TST
 (3)  026252  104432                          TRAP    C$EXIT
 (3)  026254  000012                          .WORD   L10135-.
6914  026256     000      377      377  5$:     .BYTE   0,-1,-1,376,252,-1,-1,124
      026261     376      252      377
      026264     377      124
6915
6916                                  .EVEN
6917  026266                          ENDTST
 (3)  026266                          L10135:
 (3)  026266  104401                          TRAP    C$ETST
6918
6919  026270                                  BADHEAD
 (2)                                          ;****************** TEST 52 ******************
6920                                          ;*ALU TEST
6921                                          ;*TEST OF ALU FUNCTION 2A W.C WITH C BIT CLEARED
6922                                          ;*ALU FUNCTION (A PLUS A PLUS C)        CODE-6
6923                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6924                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
6925  026270                                  BADHEAD
 (2)                                          ;****************** TEST 52 ******************
6926
6927  026270                          BGNTST
 (3)  026270                          T52::
6928  026270                                  MYINT
 (1)  026270  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
6929  026274                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  026274  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
6930  026300  005005                          CLR     R5              ;MEM + SP ADDRESS
6931  026302  012702  026462                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
6932  026306  004737  003640                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
6933  026312  002654                          MEMDAT                  ;POINTER TO DATA
6934  026314  004737  004012                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
6935  026320  002664                          SPDAT                   ;POINTER TO DATA
6936  026322                                  BGNSEG
 (3)  026322  104404                          TRAP    C$BSEG
6937  026324  004737  004060          1$:     JSR     PC,CLRC         ;CLEAR C BIT.
6938  026330  042737  000017  026346          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
6939  026336  050537  026346                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
6940  026342                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
```

```
 (1)  026342  004537  003244                JSR     R5,.ROMCLK          ;CLOCK INSTRUCTION
      026346  010000                2$:     010000                      ;LOAD MAR
      026350  042737  000017  026366        BIC     #17,3$              ;CLEAR ADDRESS OF INSTRUCTION
      026356  050537  026366                BIS     R5,3$               ;ADD ADDRESS TO INSTRUCTION
      026362                                ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  026362  004537  003244                JSR     R5,.ROMCLK          ;CLOCK INSTRUCTION
      026366  040540                3$:     040400!<6*20>               ;BR   2A W/C
      026370                                ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  026370  004537  003244                JSR     R5,.ROMCLK          ;CLOCK INSTRUCTION
      026374  061224                        61224                       ;MOVE BSR TO PORT4
      026376  111237  002636                MOVB    (R2),$GDDAT         ;PUT 'EXPECTED' IN $GDDAT
      026402  116104  000004                MOVB    4(R1),R4            ;PUT 'FOUND' IN R4
      026406  123704  002636                CMPB    $GDDAT,R4           ;DATA CORRECT?
      026412  001411                        BEQ     4$                  ;BR IF YES
      026414                                ERROR   15,YES              ;ALU ERROR
 (5)  026426  104455                        TRAP    C$ERDF
 (6)  026430  000017                        .WORD   15
 (6)  026432  004754                        .WORD   EM15
 (6)  026434  007120                        .WORD   ERR15
      026436                        4$:     ESCAPE  SEG
 (3)  026436  104410                        TRAP    C$ESCAPE
 (3)  026440  000014                        .WORD   10000$-.
      026442  005202                        INC     R2                  ;NEXT DATA
      026444  005205                        INC     R5                  ;NEXT ADDRESS
      026446  022705  000010                CMP     #10,R5              ;DONE YET?
      026452  001324                        BNE     1$                  ;BR IF NO
      026454                                ENDSEG
 (3)  026454                        10000$:
 (3)  026454  104405                        TRAP    C$ESEG
      026456                                EXIT    TST
 (3)  026456  104432                        TRAP    C$EXIT
 (3)  026460  000012                        .WORD   L10136-.
      026462     000     000     376  5$:   .BYTE   0,0,376,376,252,252,124,124
      026465     376     252     252
      026470     124     124


                                    .EVEN
      026472                        ENDTST
 (3)  026472                        L10136:
 (3)  026472  104401                        TRAP    C$ETST

      026474                                BADHEAD
 (2)                                        ;******************* TEST 53 *******************
                                            ;*ALU TEST
                                            ;*TEST OF ALU FUNCTION SUB WITH C BIT CLEARED
                                            ;*ALU FUNCTION (A-B)    CODE-16
                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
      026474                                BADHEAD
 (2)                                        ;******************* TEST 53 *******************

      026474                        BGNTST
 (3)  026474                        T53::
      026474                                MYINT
 (1)  026474  013701  002716                MOV     KMCSR,R1            ;GET DEVICE ADDRESS.
      026500                                MSTCLR                      ;MASTER CLEAR M8200,4,7
```

```
  (1)  026500  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
6976   026504  005005                          CLR     R5                  ;MEM + SP ADDRESS
6977   026506  012702  026670                  MOV     #5$,R2              ;POINTER TO CORRECT DATA
6978   026512  004737  003640                  JSR     PC,MEMLD            ;LOAD 8 WORDS OF MAIN MEMORY
6979   026516  002654                          MEMDAT                      ;POINTER TO DATA
6980   026520  004737  004012                  JSR     PC,SPLD             ;LOAD 8 WORDS OF SP
6981   026524  002664                          SPDAT                       ;POINTER TO DATA
6982   026526                                  BGNSEG
  (3)  026526  104404                          TRAP    C$BSEG
6983   026530  004737  004060          1$:     JSR     PC,CLRC             ;CLEAR C BIT!
6984   026534  042737  000017  026552          BIC     #17,2$              ;CLEAR ADDRESS FIELD OF INSTRUCTION
6985   026542  050537  026552                  BIS     R5,2$               ;ADD ADDRESS TO INSTRUCTION
6986   026546                                  ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  026546  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6987   026552  010000                  2$:     010000                      ;LOAD MAR
6988   026554  042737  000017  026572          BIC     #17,3$              ;CLEAR ADDRESS OF INSTRUCTION
6989   026562  050537  026572                  BIS     R5,3$               ;ADD ADDRESS TO INSTRUCTION
6990   026566                                  ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  026566  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6991   026572  040740                  3$:     040400!<16*20>              ;BR _ SUB
6992   026574                                  ROMCLK
  (1)  026574  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
6993   026600  061224                          61224                       ;MOVE BR TO PORT4
6994   026602  111237  002636                  MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
6995   026606  116104  000004                  MOVB    4(R1),R4            ;PUT "FOUND" IN R4
6996   026612  123737  002636  002636          CMPB    $GDDAT,$GDDAT           ;DATA CORRECT?
6997   026620  001411                          BEQ     4$                  ;BR IF YES
6998   026622                                  ERROR   15,YES              ;ALU ERROR
  (5)  026634  104455                          TRAP    C$ERDF
  (6)  026636  000017                          .WORD   15
  (6)  026640  004754                          .WORD   EM15
  (6)  026642  007120                          .WORD   ERR15
6999   026644                          4$:     ESCAPE  SEG
  (3)  026644  104410                          TRAP    C$ESCAPE
  (3)  026646  000014                          .WORD   10000$-.
7000   026650  005202                          INC     R2                  ;NEXT DATA
7001   026652  005205                          INC     R5                  ;NEXT ADDRESS
7002   026654  022705  000010                  CMP     #10,R5              ;DONE YET?
7003   026660  001323                          BNE     1$                  ;BR IF NO
7004   026662                                  ENDSEG
  (3)  026662                          10000$:
  (3)  026662  104405                          TRAP    C$ESEG
7005   026664                                  EXIT    TST
  (3)  026664  104432                          TRAP    C$EXIT
  (3)  026666  000012                          .WORD   L10137-.
7006   026670     000     001     377  5$:     .BYTE   0,1,-1,0,0,253,125,0
       026673     000     000     253
       026676     125     000
7007
7008
7009                                    .EVEN
7010   026700                           ENDTST
  (3)  026700                           L10137:
  (3)  026700  104401                          TRAP    C$ETST
7011
7012
```

```
                                        BADHEAD
                                        ;********** TEST 54 **********
                                        ;*ALU TEST
                                        ;*TEST OF ALU FUNCTION ADD W/C WITH C BIT CLEARED
                                        ;*ALU FUNCTION (A PLUS B PLUS C)        CODE=01
                                        ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                        ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
                                        BADHEAD
                                        ;********** TEST 54 **********

                                BGNTST
                                T54::
                                        MYINT
(1)  026702  013701  002716             MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
(1)  026706  004537  003156             JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
     026712  005005                     CLR     R5              ;MEM + SP ADDRESS
     026714  012702  027074             MOV     #5$,R2          ;POINTER TO CORRECT DATA
     026720  004737  003640             JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
     026724  002654                     MEMDAT                  ;POINTER TO DATA
     026726  004737  004012             JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
     026732  002664                     SPDAT                   ;POINTER TO DATA
     026734                             BGNSEG
(3)  026734  104404                     TRAP    C$BSEG
     026736  004737  004060     1$:     JSR     PC,CLRC         ;CLEAR C BIT!
     026742  042737  000017  026760     BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
     026750  050537  026760             BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
     026754                             ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
(1)  026754  004537  003244             JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
     026760  010000             2$:     010000                  ;LOAD MAR
     026762  042737  000017  027000     BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
     026770  050537  027000             BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
     026774                             ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)  026774  004537  003244             JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
     027000  040420             3$:     040400!<01*20>          ;BR   ADD W/C
     027002                             ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
(1)  027002  004537  003244             JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
     027006  061224                     61224                   ;MOVE BR TO PORT4
     027010  111237  002636             MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
     027014  116104  000004             MOVB    4(R1),R4        ;PUT "FOUND" IN R4
     027020  123704  002636             CMPB    $GDDAT,R4               ;DATA CORRECT?
     027024  001411                     BEQ     4$              ;BR IS YES
     027026                             ERROR   15,YES          ;ALU ERROR
(5)  027040  104455                     TRAP    C$ERDF
(6)  027042  000017                     .WORD   15
(6)  027044  004754                     .WORD   EM15
(6)  027046  007120                     .WORD   ERR15
     027050                     4$:     ESCAPE  SEG
(3)  027050  104410                     TRAP    C$ESCAPE
(3)  027052  000014                     .WORD   10000$-.
     027054  005202                     INC     R2              ;NEXT DATA
     027056  005205                     INC     R5              ;NEXT ADDRESS
     027060  022705  000010             CMP     #10,R5          ;DONE YET?
     027064  001324                     BNE     1$              ;BR IF NO
     027066                             ENDSEG
(3)  027066                     10000$:
```

```
  (3)  027066  104405                          TRAP    C$ESEG
7053   027070                                  EXIT    TST
  (3)  027070  104432                          TRAP    C$EXIT
  (3)  027072  000012                          .WORD   L10140-.
7054   027074     000     377     377  5$:     .BYTE   0,-1,-1,376,252,-1,-1,124
       027077     376     252     377
       027102     377     124
7055
7056                                    .EVEN
7057   027104                           ENDTST
  (3)  027104                           L10140:
  (3)  027104  104401                          TRAP    C$ETST
7058
7059
7060   027106                                  BADHEAD
  (2)                                           ;********** TEST 55 **********
7061                                            ;*ALU TEST
7062                                            ;*TEST OF ALU FUNCTION SUB W/C WITH C BIT CLEARED
7063                                            ;*ALU FUNCTION (A-B-C)  CODE=2
7064                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7065                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7066   027106                                  BADHEAD
  (2)                                           ;********** TEST 55 **********
7067
7068   027106                           BGNTST
  (3)  027106                           T55::
7069   027106                                  MYINT
  (1)  027106  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
```

```
                                                MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  027112  004537  003156                   JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
7072   027116  005005                           CLR     R5              ;MEM + SP ADDRESS
7073   027120  012702  027300                   MOV     #5$,R2          ;POINTER TO CORRECT DATA
7074   027124  004737  003640                   JSR     PC, MEMLD       ;LOAD 8 WORDS OF MAIN MEMORY
7075   027130  002654                           MEMDAT                  ;POINTER TO DATA
7076   027132  004737  004012                   JSR     PC,SPLD         ;LOAD 3 WORDS OF SP
7077   027136  002664                           SPDAT                   ;POINTER TO DATA
7078   027140                                   BGNSEG
  (3)  027140  104404                           TRAP    C$BSEG
7079   027142  004737  004060           1$:     JSR     PC,CLRC         ;CLEAR C BIT!
7080   027146  042737  000017  027164           BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7081   027154  050537  027164                   BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
7082   027160                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  027160  004537  003244                   JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7083   027164  010000                   2$:     010000                  ;LOAD MAR
7084   027166  042737  000017  027204           BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7085   027174  050537  027204                   BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
7086   027200                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
  (1)  027200  004537  003244                   JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7087   027204  040440                   3$:     040400!<2*20>           ;BR _ SUB W/C
7088   027206                                   ROMCLK
  (1)  027206  004537  003244                   JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7089   027212  061224                           61224                   ;MOVE BR TO PORT4
7090   027214  111237  002636                   MOVB    (R2),$GDDAT             ;PUT ''EXPECTED'' IN $GDDAT
7091   027220  116104  000004                   MOVB    4(R1),R4        ;PUT''FOUND'' IN R4
7092   027224  123704  002636                   CMPB    $GDDAT,R4               ;DATA CORRECT?
7093   027230  001411                           BEQ     4$              ;BR IF YES
7094   027232                                   ERROR   15,YES          ;ALU ERROR
  (5)  027244  104455                           TRAP    C$ERDF
  (6)  027246  000017                           .WORD   15
  (6)  027250  004754                           .WORD   EM15
  (6)  027252  007120                           .WORD   ERR15
7095   027254                           4$:     ESCAPE  SEG
  (3)  027254  104410                           TRAP    C$ESCAPE
  (3)  027256  000014                           .WORD   10000$-.
7096   027260  005202                           INC     R2                      ;NEXT DATA
7097   027262  005205                           INC     R5                      ;NEXT ADDRESS
7098   027264  022705  000010                   CMP     #10,R5                  ;DONE YET?
7099   027270  001324                           BNE     1$                      ;BR IF NO
7100   027272                                   ENDSEG
  (3)  027272                           10000$:
  (3)  027272  104405                           TRAP    C$ESEG
7101   027274                                   EXIT    TST
  (3)  027274  104432                           TRAP    C$EXIT
  (3)  027276  000012                           .WORD   L10141-.
7102   027300    377     000     376   5$:     .BYTE   -1,0,376,-1,-1,252,124,-1
       027303    377     377     252
       027306    124     377
7103
7104
7105                                   .EVEN
7106   027310                          ENDTST
  (3)  027310                          L10141:
  (3)  027310  104401                           TRAP    C$ETST
7107
```

```
7108
7109  027312                                    BADHEAD
 (2)                                            ;****************** TEST 56 ******************
7110                                            ;*ALU TEST
7111                                            ;*TEST OF ALU FUNCTION INC A WITH C BIT CLEARED
7112                                            ;*ALU FUNCTION (A PLUS 1)       CODE=3
7113                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7114                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7115  027312                                    BADHEAD
 (2)                                            ;****************** TEST 56 ******************
7116
7117  027312                          BGNTST
 (3)  027312                          T56::
7118  027312                                    MYINT
 (1)  027312  013701  002716                    MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7119  027316                                    MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  027316  004537  003156                    JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
7120  027322  012702  027504                    MOV     #5$,R2          ;POINTER TO CORRECT DATA
7121  027326  005005                            CLR     R5
7122  027330  004737  003640                    JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMRY
7123  027334  002654                            MEMDAT                  ;POINTER TO DATA
7124  027336  004737  004012                    JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7125  027342  002664                            SPDAT                   ;POINTER TO DATA
7126  027344                                    BGNSEG
 (3)  027344  104404                            TRAP    C$BSEG
7127  027346  004737  004060          1$:       JSR     PC,CLRC         ;CLEAR C BIT!
7128  027352  042737  000017  027370            BIC     #17,2$          ;CLEAR ADDRESS FIELD OF 'NSTRUCTION
7129  027360  050537  027370                    BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
7130  027364                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
 (1)  027364  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7131  027370  010000                  2$:       010000                  ;LOAD MAR
7132  027372  042737  000017  027410            BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7133  027400  050537  027410                    BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
7134  027404                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  027404  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7135  027410  040460                  3$:       040400!<3*20>           ;BR  INC A
7136  027412                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
 (1)  027412  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7137  027416  061224                            61224                   ;MOVE BR TO PORT4
7138  027420  111237  002636                    MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
7139  027424  116104  000004                    MOVB    4(R1),R4        ;PUT "FOUND" IN R4
7140  027430  123704  002636                    CMPB    $GDDAT,R4               ;DATA CORRECT?
7141  027434  001411                            BEQ     4$              ;BR IF YES
7142  027436                                    ERROR   15,YES          ;ALU ERROR
 (5)  027450  104455                            TRAP    C$ERDF
 (6)  027452  000017                            .WORD   15
 (6)  027454  004754                            .WORD   EM15
 (6)  027456  007120                            .WORD   ERR15
7143  027460                          4$:       ESCAPE  SEG
 (3)  027460  104410                            TRAP    C$ESCAPE
 (3)  027462  000014                            .WORD   10000$-.
7144  027464  005202                            INC     R2              ;NEXT DATA
7145  027466  005205                            INC     R5
7146  027470  022705  000010                    CMP     #10,R5          ;DONE YET?
7147  027474  001324                            BNE     1$              ;BR IF NO
7148  027476                                    ENDSEG
```

```
 (3) 027476                                  10000$:
 (3) 027476  104405                                  TRAP    C$ESEG
7149 027500                                          EXIT    TST
 (3) 027500  104432                                  TRAP    C$EXIT
 (3) 027502  000012                                  .WORD   L10142-.
7150 027504     001     001     000  5$:     .BYTE   1,1,0,0,126,126,253,253
     027507     000     126     126
     027512     253     253
7151
7152                                         .EVEN
7153 027514                                  ENDTST
 (3) 027514                                  L10142:
 (3) 027514  104401                                  TRAP    C$ETST
7154
7155
7156 027516                                          BADHEAD
 (2)                                                 ;****************** TEST 57 ******************
7157                                                 ;*ALU TEST
7158                                                 ;*TEST OF ALU FUNCTION 2A WITH C BIT CLEARED
7159                                                 ;*ALU FUNCTION (A PLUS A)       CODE=5
7160                                                 ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7161                                                 ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7162 027516                                          BADHEAD
 (2)                                                 ;****************** TEST 57 ******************
7163
7164 027516                                  BGNTST
 (3) 027516                                  T57::
7165 027516                                          MYINT
 (1) 027516  013701  002716                          MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7166 027522                                          MSTCLR                  ;MASTER CLEAR DMC11
 (1) 027522  004537  003156                          JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
7167 027526  005005                                  CLR     R5              ;MEM * SP ADDRESS
7168 027530  012702  027710                          MOV     #5$,.2          ;POINTER TO CORRECT DATA
7169 027534  004737  003640                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7170 027540  002654                                  MEMDAT                  ;POINTER TO DATA
7171 027542  004737  004012                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7172 027546  002664                                  SPDAT                   ;POINTER TO DATA
7173 027550                                          BGNSEG
 (3) 027550  104404                                  TRAP    C$BSEG
7174 027552  004737  004060              1$:         JSR     PC,CLRC         ;CLEAR C BIT!
7175 027556  042737  000017  027574                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7176 027564  050537  027574                          BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
7177 027570                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1) 027570  004537  003244                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7178 027574  010000                      2$:         010000                  ;LOAD MAR
7179 027576  042737  000017  027614                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7180 027604  050537  027614                          BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
7181 027610                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1) 027610  004537  003244                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7182 027614  040520                      3$:         040400!<5*20>           ;BR   2A
7183 027616                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1) 027616  004537  003244                          JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7184 027622  061224                                  61224                   ;MOVE BR TO PORT4
7185 027624  111237  002636                          MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
7186 027630  116104  000004                          MOVB    4(R1),R4        ;PUT "FOUND" IN R4
7187 027634  123704  002636                          CMPB    $GDDAT,R4               ;DATA CORRECT?
```

```
7188  027640  001411                          BEQ     4$              ;BR IF YES
7189  027642                                  ERROR   15,YES          ;ALU ERROR
 (5)  027654  104455                          TRAP    C$ERDF
 (6)  027656  000017                          .WORD   15
 (6)  027660  004754                          .WORD   EM15
 (6)  027662  007120                          .WORD   ERR15
7190  027664                            4$:   ESCAPE  SEG
 (3)  027664  104410                          TRAP    C$ESCAPE
 (3)  027666  000014                          .WORD   10000$-.
7191  027670  005202                          INC     R2              ;NEXT DATA
7192  027672  005205                          INC     R5              ;NEXT ADDRESS
7193  027674  022705  000010                  CMP     #10,R5          ;DONE YET?
7194  027700  001324                          BNE     1$              ;BR IF NO
7195  027702                                  ENDSEG
 (3)  027702                            10000$:
 (3)  027702  104405                          TRAP    C$ESEG
7196  027704                                  EXIT    TST
 (3)  027704  104432                          TRAP    C$EXIT
 (3)  027706  000012                          .WORD   L10143-.
7197  027710     000     000     376   5$:    .BYTE   0,0,376,376,252,252,124,124
      027713     376     252     252
      027716     124     124
7198
7199                                    .EVEN
7200  027720                            ENDTST
 (3)  027720                            L10143:
 (3)  027720  104401                          TRAP    C$ETST
7201
7202
7203  027722                                  BADHEAD
 (2)                                          ;****************** TEST 58 ******************
7204                                          ;*ALU TEST
7205                                          ;*TEST OF ALU FUNCTION A PLUS C WITH C BIT CLEARED
7206                                          ;*ALU FUNCTION (A PLUS C)       CODE-4
7207                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7208                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7209  027722                                  BADHEAD
 (2)                                          ;****************** TEST 58 ******************
7210
7211  027722                            BGNTST
 (3)  027722                            T58::
7212  027722                                  MYINT
 (1)  027722  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7213  027726                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  027726  004537  003156                  JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
7214  027732  005005                          CLR     R5              ;MEM + SP ADDRESS
7215  027734  012702  030114                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
7216  027740  004737  003640                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7217  027744  002654                          MEMDAT                  ;POINTER TO DATA
7218  027746  004737  004012                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7219  027752  002664                          SPDAT                   ;POINTER TO DATA
7220  027754                                  BGNSEG
 (3)  027754  104404                          TRAP    C$BSEG
7221  027756  004737  004060            1$:   JSR     PC,CLRC         ;CLEAR C BIT!
7222  027762  042737  000017  030000         BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7223  027770  050537  030000                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
```

```
7224  027774                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
 (1)  027774  004537  003244                        JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7225  030000  010000                        2$:     010000                  ;LOAD MAR
7226  030002  042737  000017  030020              BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
7227  030010  050537  030020                      BIS     R5,3$                   ;ADD ADDRESS TO INSTRUCTION
7228  030014                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=304
 (1)  030014  004537  003244                        JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7229  030020  040500                        3$:     040400.<4*20>           ;BR   A PLUS C
7230  030022                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCKL PC=5305
 (1)  030022  004537  003244                        JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7231  030026  061224                                61224                   ;MOVE BR TO PORT4
7232  030030  111237  002636                        MOVB    (R2),$GDDAT             ;PUT 'EXPECTED" IN $GDDAT
7233  030034  116104  000004                        MOVB    4(R1),R4        ;PUT 'FOUND" IN R4
7234  030040  123704  002636                        CMPB    $GDDAT,R4               ;DATA CORRECT?
7235  030044  001411                                BEQ     4$              ;BR IS YES
7236  030046                                        ERROR   15,YES          ;ALU ERROR
 (5)  030060  104455                                TRAP    C$ERDF
 (6)  030062  000017                                .WORD   15
 (6)  030064  004754                                .WORD   EM15
 (6)  030066  007120                                .WORD   ERR15
7237  030070                                4$:     ESCAPE  SEG
 (3)  030070  104410                                TRAP    C$ESCAPE
 (3)  030072  000014                                .WORD   10000$-.
7238  030074  005202                                INC     R2              ;NEXT DATA
7239  030076  005205                                INC     R5              ;NEXT ADDRESS
7240  030100  022705  000010                        CMP     #10,R5          ;DONE YET?
7241  030104  001324                                BNE     1$              ;BR IF NO
7242  030106                                        ENDSEG
 (3)  030106                            10000$:
 (3)  030106  104405                                TRAP    C$ESEG
7243  030110                                        EXIT    TST
 (3)  030110  104432                                TRAP    C$EXIT
 (3)  030112  000012                                .WORD   L10144-.
7244  030114     000     000     377  5$:     .BYTE   0,0,-1,-1,125,125,252,252
      030117     377     125     125
      030122     252     252
7245
7246                                          .EVEN
7247  030124                                  ENDTST
 (3)  030124                              L10144:
 (3)  030124  104401                                TRAP    C$ETST
7248
7249
7250  030126                                        BADHEAD
 (2)                                              ;****************** TEST 59 ******************
7251                                              ;*ALU TEST
7252                                              ;*TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT CLEARED
7253                                              ;*ALU FUNCTION (A-B-1)  CODE-17
7254                                              ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7255                                              ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7256  030126                                      BADHEAD
 (2)                                              ;****************** TEST 59 ******************
7257
7258  030126                                  BGNTST
 (3)  030126                              T59::
7259  030126                                        MYINT
```

```
  (1)  030126  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7260  030132                                MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  030132  004537  003156                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 7261  030136  005005                        CLR     R5              ;MEM + SP ADDRESS
 7262  030140  012702  030320                MOV     #5$,R2          ;POINTER TO CORRECT DATA
 7263  030144  004737  003640                JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 7264  030150  002654                        MEMDAT                  ;POINTER TO DATA
 7265  030152  004737  004012                JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 7266  030156  002664                        SPDAT                   ;POINTER TO DATA
 7267  030160                                BGNSEG
  (3)  030160  104404                        TRAP    C$BSEG
 7268  030162  004737  004060        1$:     JSR     PC,CLRC         ;CLEAR C BIT!
 7269  030166  042737  000017  030204        BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7270  030174  050537  030204                BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 7271  030200                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  030200  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7272  030204  010000                2$:     010000                  ;LOAD MAR
 7273  030206  042737  000017  030224        BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 7274  030214  050537  030224                BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 7275  030220                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
  (1)  030220  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7276  030224  040760                3$:     040400!<17*20>          ;BR   2'S COMP SUB
 7277  030226                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  030226  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7278  030232  061224                        61224                   ;MOVE BR TO PORT4
 7279  030234  111237  002636                MOVB    (R2),$GDDAT             ;PUT 'EXPECTED' IN $GDDAT
 7280  030240  116104  000004                MOVB    4(R1),R4        ;PUT 'FOUND' IN R4
 7281  030244  123704  002636                CMPB    $GDDAT,R4               ;DATA CORRECT?
 7282  030250  001411                        BEQ     4$              ;BR IS YES
 7283  030252                                ERROR   15,YES          ;ALU ERROR
  (5)  030264  104455                        TRAP    C$ERDF
  (6)  030266  000017                        .WORD   15
  (6)  030270  004754                        .WORD   EM15
  (6)  030272  007120                        .WORD   ERR15
 7284  030274                        4$:     ESCAPE  SEG
  (3)  030274  104410                        TRAP    C$ESCAPE
  (3)  030276  000014                        .WORD   10000$-.
 7285  030300  005202                        INC     R2              ;NEXT DATA
 7286  030302  005205                        INC     R5              ;NEXT ADDRESS
 7287  030304  022705  000010                CMP     #10,R5          ;DONE YET?
 7288  030310  001324                        BNE     1$              ;BR IF NO
 7289  030312                                ENDSEG
  (3)  030312                        10000$:
  (3)  030312  104405                        TRAP    C$ESEG
 7290  030314                                EXIT    TST
  (3)  030314  104432                        TRAP    C$EXIT
  (3)  030316  000012                        .WORD   L10145-.
 7291  030320     377     000     376  5$:     .BYTE   -1,0,376,-1,-1,252,124,-1
       030323     377     377     252
       030326     124     377
 7292
 7293                                .EVEN
 7294  030330                        ENDTST
  (3)  030330                        L10145:
  (3)  030330  104401                        TRAP    C$ETST
 7295
```

```
7296
7297  030332                                    BADHEAD
 (2)                                            ;****************** TEST 60 ******************
7298                                            ;*ALU TEST
7299                                            ;*TEST OF ALU FUNCTION DEC A WITH C BIT CLEARED
7300                                            ;*ALU FUNCTION (A-1)    CODE=7
7301                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7302                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7303
7304  030332                                    BADHEAD
 (2)                                            ;****************** TEST 60 ******************
7305
7306  030332                          BGNTST
 (3)  030332                          T60::
7307  030332                                    MYINT
 (1)  030332  013701  002716                    MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7308  030336                                    MSTCLR                  ;MASTER CLEAR DMC11
 (1)  030336  004537  003156                    JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
7309  030342  005005                           CLR     R5              ;MEM + SP ADDRESS
7310  030344  012702  030524                   MOV     #5$,R2          ;POINTER TO CORRECT DATA
7311  030350  004737  003640                   JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMMOR
7312  030354  002654                           MEMDAT                  ;POINTER TO DATA
7313  030356  004737  004012                   JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7314  030362  002664                           SPDAT                   ;POINTER TO DATA
7315  030364                                    BGNSEG
 (3)  030364  104404                           TRAP    C$BSEG
7316  030366  004737  004060          1$:     JSR     PC,CLRC         ;CLEAR C BIT.
7317  030372  042737  000017  030410           BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7318  030400  050537  030410                   BIS     R5,2$           ;ADD ADDRESS TO INSTRUCITON
7319  030404                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
 (1)  030404  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7320  030410  010000                  2$:     010000                  ;LOAD MAR
7321  030412  042737  000017  030430           BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7322  030420  050537  030430                   BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
7323  030424                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
 (1)  030424  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7324  030430  040560                  3$:     040400!<7*20>           ;BR   DEC A
7325  030432                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
 (1)  030432  004537  003244                    JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7326  030436  061224                           61224                   ;MOVE BR TO PORT4
7327  030440  111237  002636                   MOVB    (R2),$GDDAT             ;PUT 'EXPECTED'' IN $GDDAT
7328  030444  116104  000004                   MOVB    4(R1),R4        ;PUT 'FOUND'' IN R4
7329  030450  123704  002636                   CMPB    $GDDAT,R4               ;DATA CORRECT?
7330  030454  001411                           BEQ     4$              ;BR IF YE
7331  030456                                    ERROR   15,YES          ;ALU ERROR
 (5)  030470  104455                           TRAP    C$ERDF
 (6)  030472  000017                           .WORD   15
 (6)  030474  004754                           .WORD   EM15
 (6)  030476  007120                           .WORD   ERR15
7332  030500                          4$:     ESCAPE  SEG
 (3)  030500  104410                           TRAP    C$ESCAPE
 (3)  030502  000014                           .WORD   10000$-.
7333  030504  005202                           INC     R2              ;NEXT DATA
7334  030506  005205                           INC      R5             ;NEXT ADDRESS
7335  030510  022705  000010                   CMP     #10,R5          ;DONE YET?
7336  030514  001324                           BNE      1$             ;BR IF NO
```

```
 7337  030516                                    ENDSEG
  (3)  030516                              10000$:
  (3)  030516  104405                             TRAP    C$ESEG
 7338  030520                                    EXIT    TST
  (3)  030520  104432                             TRAP    C$EXIT
  (3)  030522  000012                             .WORD   L10146-.
 7339  030524     377     377     376  5$:       .BYTE   -1,-1,376,376,124,124,251,251
       030527     376     124     124
       030532     251     251
 7340
 7341                                    .EVEN
 7342  030534                            ENDTST
  (3)  030534                            L10146:
  (3)  030534  104401                             TRAP    C$ETST
 7343
 7344
 7345  030536                                    BADHEAD
  (2)                                            ;****************** TEST 61 ******************
 7346                                            ;*ALU TEST
 7347                                            ;*TEST OF ALU FUNCTION SEL B WITH C BIT SET
 7348                                            ;*ALU FUNCTION (B)      CODE=11
 7349                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 7350                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 7351  030536                                    BADHEAD
  (2)                                            ;****************** TEST 61 ******************
 7352
 7353  030536                            BGNTST
  (3)  030536                            T61::
 7354  030536                                    MYINT
  (1)  030536  013701  002716                     MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7355  030542                                    MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  030542  004537  003156                     JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 7356  030546  005005                             CLR     R5              ;MEM + SP ADDRESS
 7357  030550  012702  030730                     MOV     #5$,R2          ;POINTER TO CORRECT DATA
 7358  030554  004737  003640                     JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 7359  030560  002654                             MEMDAT                  ;POINTER TO DATA
 7360  030562  004737  004012                     JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 7361  030566  002664                             SPDAT                   ;POINTET TO DATA
 7362  030570                                    BGNSEG
  (3)  030570  104404                             TRAP    C$BSEG
 7363  030572  004737  004076          1$:       JSR     PC,SETC         ;SET C BIT!
 7364  030576  042737  000017  030614             BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7365  030604  050537  030614                     BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 7366  030610                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
  (1)  030610  00  37  003244                     JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7367  030614  010000                    2$:     010000                  ;LOAD MAR
 7368  030616  042737  000017  030634             BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 7369  030624  050537  030634                     BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 7370  030630                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  030630  004537  003244                     JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7371  030634  040620                    3$:     040400.<11*20>          ;BR   SEL B
 7372  030636                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  030636  004537  003244                     JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7373  030642  061224                             61224                   ;MOVE BR TO PORT4
 7374  030644  111237  002636                     MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
 7375  030650  116104  000004                     MOVB    4(R1),R4        ;PUT "FOUND" IN R4
```

```
7376  030654  123704  002656          CMPB    $GDDAT,R4               ;DATA CORRECT?
7377  030660  001411                  BEQ     4$                      ;BR IF YES
7378  030662                          ERROR   23,YES                  ;ALU ERROR
 (5)  030674  104455                  TRAP    C$ERDF
 (6)  030676  000027                  .WORD   23
 (6)  030700  005226                  .WORD   EM23
 (6)  030702  007404                  .WORD   ERR23
7379  030704                  4$:     ESCAPE  SEG
 (3)  030704  104410                  TRAP    C$ESCAPE
 (3)  030706  000014                  .WORD   10000$-.
7380  030710  005202                  INC     R2                      ;NEXT DATA
7381  030712  005205                  INC     R5                      ;NEXT ADDRESS
7382  030714  022705  000010          CMP     #10,R5                  ;DONE YET?
7383  030720  001324                  BNE     1$                      ;BR IF NO
7384  030722                          ENDSEG
 (3)  030722                  10000$:
 (3)  030722  104405                  TRAP    C$ESEG
7385  030724                          EXIT    TST
 (3)  030724  104432                  TRAP    C$EXIT
 (3)  030726  000012                  .WORD   L10147-.
7386  030730    000     377     000   5$:     .BYTE   0,-1,0,-1,125,252,125,252
      030733    377     125     252
      030736    125     252
7387
7388                                  .EVEN
7389  030740                          ENDTST
 (3)  030740                  L10147:
 (3)  030740  104401                  TRAP    C$ETST
7390
7391
7392  030742                          BADHEAD
 (2)                                  ;******************** TEST 62 ********************
7393                                  ;*ALU TEST
7394                                  ;*TEST OF ALU FUNCTION SEL A WITH C BIT SET
7395                                  ;*ALU FUNCTION (A)      CODE=10
7396                                  ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7397                                  ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7398  030742                          BADHEAD
 (2)                                  ;******************** TEST 62 ********************
```

```
7400
7401
7402  030742                          BGNTST
 (3)  030742                          T62::
7403  030742                                MYINT
 (1)  030742  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7404  030746                                MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  030746  004537  003156                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
7405  030752  005005                       CLR     R5              ;MEM + SP ADDRESS
7406  030754  012702  031134               MOV     #5$,R2          ;POINTER TO CORRECT DATA
7407  030760  004737  003640               JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7408  030764  002654                       MEMDAT                  ;POINTER TO DATA
7409  030766  004737  004012               JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7410  030772  002664                       SPDAT                   ;POINTER TO DATA
7411  030774                                BGNSEG
 (3)  030774  104404                       TRAP    C$BSEG
7412  030776  004737  004076        1$:    JSR     PC,SETC         ;SET C BIT!
7413  031002  042737  000017  031020       BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7414  031010  050537  031020               BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
7415  031014                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
 (1)  031014  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7416  031020  010000                2$:    010000                  ;LOAD MAR
7417  031022  042737  000017  031040       BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCITON
7418  031030  050537  031040               BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
7419  031034                                ROMCLK                  ;NEXT WORD IS INSTRUCITON, ROMCLK PC-5304
 (1)  031034  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7420  031040  040600                3$:    040400.<10*20>          ;BR   SEL A
7421  031042                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
 (1)  031042  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7422  031046  061224                       61224                   ;MOVE BR TO PORT4
7423  031050  111237  002636               MOVB    (R2),$GDDAT             ;PUT 'EXPECTED'' IN $GDDAT
7424  031054  116104  000004               MOVB    4(R1),R4        ;PUT 'FOUND'' IN R4
7425  031060  123704  002636               CMPB    $GDDAT,R4               ;DATA CORRECT?
7426  031064  001411                       BEQ     4$              ;BR IF YES
7427  031066                                ERROR   23,YES          ;ALU ERROR
 (5)  031100  104455                       TRAP    C$ERDF
 (6)  031102  000027                       .WORD   23
 (6)  031104  005226                       .WORD   EM23
 (6)  031106  007404                       .WORD   ERR23
7428  031110                          4$:   ESCAPE  SEG
 (3)  031110  104410                       TRAP    C$ESCAPE
 (3)  031112  000014                       .WORD   10000$-.
7429  031114  005202                       INC     R2              ;NEXT DATA
7430  031116  005205                       INC     R5              ;NEXT ADDRESS
7431  031120  022705  000010               CMP     #10,R5          ;DONE YET?
7432  031124  001324                       BNE     1$              ;BR IF NO
7433  031126                                ENDSEG
 (3)  031126                          10000$:
 (3)  031126   04405                       TRAP    C$ESEG
7434  031130                                EXIT    TST
 (3)  031130  104432                       TRAP    C$EXIT
 (3)  031132  000012                       .WORD   L10150-.
7435  031134     000       000       377  5$:    .BYTE   0,0,-1,-1,125,125,252,252
      031137     377       125       125
      031142     252       252
7436
```

```
                                        .EVEN
                                        ENDTST
                                        L10150:
                                                TRAP    C$ETST

                                                BADHEAD
                                                ;****************** TEST 63 ******************
                                                ;*ALU TEST
                                                ;*TEST OF ALU FUNCTION A OR NOTB WITH C BIT SET
                                                ;*ALU FUNCTION (A OR NOTB)      CODE=12
                                                ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
                                                BADHEAD
                                                ;****************** TEST 63 ******************

                                        BGNTST
                                        T63::
                                                MYINT
                                                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
                                                MSTCLR                  ;MASTER CLEAR M8200,4,7
                                                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
                                                CLR     R5              ;MEM + SP ADDRESS
                                                MOV     #5$,R2          ;POINTER TO CORRECT DATA
                                                JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
                                                MEMDAT                  ;POINTER TO DATA
                                                JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
                                                SPDAT                   ;POINTER TO DATA
                                                BGNSEG
                                                TRAP    C$BSEG
                                        1$:     JSR     PC,SETC         ;SET C BIT.
                                                BIC     #17,2$          ;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$:     040400!<12*20>          ;BR   A OR NOTB
                                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
                                                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
                                                61224                   ;MOVE BR TO PORT4
                                                MOVB    (R2),$GDDAT             ;PUT ''EXPECTED'' IN $GDDAT
                                                MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
                                                CMPB    $GDDAT,R4               ;DATA CORRECT?
                                                BEQ     4$              ;BR IF YES
                                                ERROR    15,YES         ;ALU ERROR
                                                TRAP    C$ERDF
                                                .WORD   15
                                                .WORD   EM15
                                                .WORD   ERR15
                                        4$:     ESCAPE  SEG
                                                TRAP    C$ESCAPE
                                                .WORD   10000$-.
```

```
7476  031320  005202                          INC     R2              ;NEXT DATA
7477  031322  005205                          INC     R5              ;NEXT ADDRESS
7478  031324  022705  000010                  CMP     #10,R5          ;DONE YET?
7479  031330  001324                          BNE     1$              ;BR IF NO
7480  031332                                  ENDSEG
 (3)  031332                          10000$:
 (3)  031332  104405                          TRAP    C$ESEG
7481  031334                                  EXIT    TST
 (3)  031334  104432                          TRAP    C$EXIT
 (3)  031336  000012                          .WORD   L10151-.
7482  031340     377     000     377  5$:     .BYTE   -1,0,-1,-1,-1,125,252,-1
      031343     377     377     125
      031346     252     377
7483
7484                                  .EVEN
7485  031350                          ENDTST
 (3)  031350                          L10151:
 (3)  031350  10440'                          TRAP    C$ETST
7486
7487
7488  031352                                  BADHEAD
 (2)                                          ;****************** TEST 64 ******************
7489                                          ;*ALU TEST
7490                                          ;*TEST OF ALU FUNCTION A AND B WITH C BIT SET
7491                                          ;ALU FUNCTION (A AND B) CODE=13
7492                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7493                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7494  031352                                  BADHEAD
 (2)                                          ;****************** TEST 64 ******************
7495
7496  031352                          BGNTST
 (3)  031352                          T64::
7497  031352                                  MYINT
 (1)  031352  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7498  031356                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  031356  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
7499  031362  005005                          CLR     R5              ;MEM + SP ADDRESS
7500  031364  012702  031544                  MOV     #5$,R2          ;POINTER TO CORRECT ADDRESS
7501  031370  004737  003640                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7502  031374  002654                          MEMDAT                  ;POINTER TO DATA
7503  031376  004737  004012                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7504  031402  002664                          SPDAT                   ;POINTER TO DATA
7505  031404                                  BGNSEG
 (3)  031404  104404                          TRAP    C$BSEG
7506  031406  004737  004076          1$:     JSR     PC,SETC         ;SET C BIT!
7507  031412  042737  000017  031430          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7508  031420  050537  031430                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
7509  031424                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  031424  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7510  031430  010000                  2$:     010000                  ;LOAD MAR
7511  031432  042737  000017  031450          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7512  031440  050537  031450                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCITON
7513  031444                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  031444  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7514  031450  040660                  3$:     040400.<13*20>          ;BR   A AND B
7515  031452                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
```

```
  (1)  031452  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7516   031456  061224                          61224                   ;MOVE BR TO PORT4
7517   031460  111237  002636                  MOVB    (R2),$GDDAT             ;PUT 'EXPECTED' IN $GDDAT
7518   031464  116104  000004                  MOVB    4(R1),R4        ;PUT 'FOUND' IN R4
7519   031470  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
7520   031474  001411                          BEQ     4$              ;BR IF YES
7521   031476                                  ERROR   23,YES          ;ALU ERROR
  (5)  031510  104455                          TRAP    C$ERDF
  (6)  031512  000027                          .WORD   23
  (6)  031514  005226                          .WORD   EM23
  (6)  031516  007404                          .WORD   ERR23
7522   031520                          4$:     ESCAPE  SEG
  (3)  031520  104410                          TRAP    C$ESCAPE
  (3)  031522  000014                          .WORD   10000$-.
7523   031524  005202                          INC     R2              ;NEXT DATA
7524   031526  005205                          INC     R5              ;NEXT ADDRESS
7525   031530  022705  000010                  CMP     #10,R5          ;DONE YET?
7526   031534  001324                          BNE     1$              ;BR IF NO
7527   031536                                  ENDSEG
  (3)  031536                          10000$:
  (3)  031536  104405                          TRAP    C$ESEG
7528   031540                                  EXIT    TST
  (3)  031540  104432                          TRAP    C$EXIT
  (3)  031542  000012                          .WORD   L10152-.
7529   031544     000      000      000  5$:     .BYTE   0,0,0,-1,125,0,0,252
       031547     377      125      000
       031552     000      252
7530
7531                                    .EVEN
7532   031554                          ENDTST
  (3)  031554                          L10152:
  (3)  031554  104401                          TRAP    C$ETST
7533
7534
7535   031556                                  BADHEAD
  (2)                                           ;********** TEST 65 **********
7536                                            ;*ALU TEST
7537                                            ;*TEST OF ALU FUNCTION A OR B WITH C BIT SET
7538                                            ;*ALU FUNCTION (A OR B) CODE=14
7539                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7540                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7541   031556                                   BADHEAD
  (2)                                           ;********** TEST 65 **********
7542
7543   031556                          BGNTST
  (3)  031556                          T65::
7544   031556                                  MYINT
  (1)  031556  013701  002716                  MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7545   031562                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  031562  004537  003156                  JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
7546   031566  005005                          CLR     R5              ;MEM + SP ADDRESS
7547   031570  012702  031750                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
7548   031574  004737  003640                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7549   031600  002654                          MEMDAT                  ;POINTER TO DATA
7550   031602  004737  004012                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7551   031606  002664                          SPDAT                   ;POINTER TO DATA
```

```
 7552  031610                                          BGNSEG
  (3)  031610  104404                                  TRAP    C$BSEG
 7553  031612  004737  004076              1$:       JSR     PC,SETC         ;SET C BIT!
 7554  031616  042737  000017  031634                BIC #17,2$              ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7555  031624  050537  031634                        BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 7556  031630                                       ROMCLK                  ;NEXT WORD IS INSTRUCITON, ROMCLK PC=5304
  (1)  031630  004537  003244                        JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7557  031634  010000                       2$:     010000                  ;LOAD MAR
 7558  031636  042737  000017  031654              BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 7559  031644  050537  031654                        BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 7560  031650                                       ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  031650  004537  003244                        JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7561  031654  040700                       3$:     040400.<14*20>          ;BR   A OR B
 7562  031656                                       ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  031656  004537  003244                        JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7563  031662  061224                               61224                   ;MOVE BR TO PORT4
 7564  031664  111237  002636                       MOVB    (R2),$GDDAT             ;PUT 'EXPECTED'IN R4
 7565  031670  116104  000004                       MOVB    4(R1),R4        ;PUT 'FOUND' IN R4
 7566  031674  123704  002636                       CMPB    $GDDAT,R4               ;DATA CORRECT?
 7567  031700  001411                               BEQ     4$              ;BR IF YES
 7568  031702                                       ERROR   23,YES          ;ALU ERROR
  (5)  031714  104455                               TRAP    C$ERDF
  (6)  031716  000027                               .WORD   23
  (6)  031720  005226                               .WORD   EM23
  (6)  031722  007404                               .WORD   ERR23
 7569  031724                               4$:     ESCAPE  SEG
  (3)  031724  104410                               TRAP    C$ESCAPE
  (3)  031726  000014                               .WORD   10000$-.
 7570  031730  005202                               INC     R2              ;NEXT DATA
 7571  031732  005205                               INC     R5              ;NEXT ADDRESS
 7572  031734  022705  000010                       CMP     #10,R5          ;DONE YET?
 7573  031740  001324                               BNE     1$              ;BR IF NO
 7574  031742                                       ENDSEG
  (3)  031742                           10000$:
  (3)  031742  104405                               TRAP    C$ESEG
 7575  031744                                       EXIT    TST
  (3)  031744  104432                               TRAP    C$EXIT
  (3)  031746  000012                               .WORD   L10153-.
 7576  031750     000      377      377  5$:       .BYTE   0,-1,-1,-1,125,-1,-1,252
       031753     377      125      377
       031756     377      252
 7577
 7578                                   .EVEN
 7579  031760                           ENDTST
  (3)  031760                           L10153:
  (3)  031760  104401                               TRAP    C$ETST
 7580
 7581
 7582  031762                                       BADHEAD
  (2)                                              ;******************** TEST 66 ********************
 7583                                              ;*ALU TEST
 7584                                              ;*TEST OF ALU FUNCTION A XOR B WITH C BIT SET
 7585                                              ;*ALU FUNCTION (A XOR B)          CODE=15
 7586                                              ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 7587                                                      ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 7588
```

```
 7589  031762                                          BADHEAD
  (2)                                                ;****************** TEST 66 ******************
 7590  031762                                  BGNTST
  (3)  031762                                  T66::
 7591  031762                                          MYINT
  (1)  031762  013701  002716                         MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7592  031766                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  031766  004537  003156                         JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
 7593  031772  005005                                 CLR     R5              ;MEM + SP ADDRESS
 7594  031774  012702  032154                         MOV     #5$,R2          ;POINTER TO CORRECT DATA
 7595  032000  004737  003640                         JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
 7596  032004  002654                                 MEMDAT                  ;POINTER TO DATA
 7597  032006  004737  004012                         JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 7598  032012  002664                                 SPDAT                   ;POINTER TO DATA
 7599  032014                                          BGNSEG
  (3)  032014  104404                                 TRAP    C$BSEG
 7600  032016  004737  004076                 1$:     JSR     PC,SETC         ;SET C BIT!
 7601  032022  042737  000017  032040                 BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7602  032030  050537  032040                         BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 7603  032034                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  032034  004537  003244                         JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7604  032040  010000                          2$:     010000                  ;LOAD MAR
 7605  032042  042737  000017  032060                 BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 7606  032050  050537  032060                         BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 7607  032054                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
  (1)  032054  004537  003244                         JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7608  032060  040720                          3$:     040400!<15*20>          ;BR   A XOR B
 7609  032062                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  032062  004537  003244                         JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
 7610  032066  061224                                 61224                   ;MOVE BR TO PORT4
 7611  032070  111237  002636                         MOVB    (R2),$GDDAT             ;PUT 'EXPECTED" IN $GDDAT
 7612  032074  116104  000004                         MOVB    4(R1),R4        ;PUT 'FOUND" IN R4
 7613  032100  123704  002636                         CMPB    $GDDAT,R4               ;DATA CORRECT?
 7614  032104  001411                                 BEQ     4$              ;BR IF YES
 7615  032106                                          ERROR   23,YES          ;ALU ERROR
  (5)  032120  104455                                 TRAP    C$ERDF
  (6)  032122  000027                                 .WORD   23
  (6)  032124  005226                                 .WORD   EM23
  (6)  032126  007404                                 .WORD   ERR23
 7616  032130                                  4$:     ESCAPE  SEG
  (3)  032130  104410                                 TRAP    C$ESCAPE
  (3)  032132  000014                                 .WORD   10000$-.
 7617  032134  005202                                 INC     R2              ;NEXT DATA
 7618  032136  005205                                 INC     R5              ;NEXT ADDRESS
 7619  032140  022705  000010                         CMP     #10,R5          ;DONE YET?
 7620  032144  001324                                 BNE     1$              ;BR IF NO
 7621  032146                                          ENDSEG
  (3)  032146                              10000$:
  (3)  032146  104405                                 TRAP    C$ESEG
 7622  032150                                          EXIT    TST
  (3)  032150  104432                                 TRAP    C$EXIT
  (3)  032152  000012                                 .WORD   L10154-.
 7623  032154     000      377      377  5$:     .BYTE   0,-1,-1,0,0,-1,-1,0
       032157     000      000      377
       032162     377      000
 7624
```

```
 7625                                   .EVEN
 7626  032164                           ENDTST
  (2)  032164                           L10154:
  (3)  032164  104401                           TRAP    C$ETST
 7627
 7628
 7629  032166                                   BADHEAD
  (2)                                           ;****************** TEST 67 ******************
 7630                                           ;*ALU TEST
 763                                            ;*TEST OF ALU FUNCTION ADD WITH C BIT SET
 7632                                           ;*ALU FUNCTION (A PLUS B)       CODE=00
 7633                                           ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 7634                                           ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 7635  032166                                   BADHEAD
  (2)                                           ;****************** TEST 67 ******************
 7636
 7637  032166                           BGNTST
  (3)  032166                           T67::
 7638  032166                                   MYINT
  (1)  032166  013701  002716                   MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7639  032172                                   MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  032172  004537  003156                   JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 7640  032176  005005                           CLR     R5              ;MEM + SP ADDRESS
 7641  032200  012702  032360                   MOV     #5$,R2          ;POINTER TO CORRECT DATA
 7642  032204  004737  003640                   JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 7643  032210  002654                           MEMDAT                  ;POINTER TO DATA
 7644  032212  004737  004012                   JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 7645  032216  002664                           SPDAT                   ;OOINTER TO DATA
 7646  032220                                   BGNSEG
  (3)  032220  104404                           TRAP    C$BSEG
 7647  032222  004737  004076           1$:     JSR     PC,SETC         ;SET C BIT!
 7648  032226  042737  000017  032244           BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7649  032234  050537  032244                   BIS     R5,2$           ;ADD ADDRESS TO INSTRUCITON
 7650  032240                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  032240  004537  003244                   JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7651  032244  010000                   2$:     010000                  ;LOAD MAR
 7652  032246  042737  000017  032264           BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 7653  032254  050537  032264                   BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 7654  032260                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  032260  004537  003244                   JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7655  032264  040400                   3$:     040400!<00*20>          ;BR   ADD
 7656  032266                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
  (1)  032266  004537  003244                   JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7657  032272  061224                           61224                   ;MOVE BR TO PORT4
 7658  032274  111237  002636                   MOVB    (R2),$GDDAT             ;PUT ''EXPECTED'' IN $GDDAT
 7659  032300  116104  000004                   MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
 7660  032304  123704  002636                   CMPB    $GDDAT,R4               ;DATA CORRECT?
 7661  032310  001411                           BEQ     4$              ;BR IF YES
 7662  032312                                   ERROR   23,YES          ;ALU ERROR
  (5)  032324  104455                           TRAP    C$ERDF
  (6)  032326  000027                           .WORD   23
  (6)  032330  005226                           .WORD   EM23
  (6)  032332  007404                           .WORD   ERR23
 7663  032334                           4$:     ESCAPE  SEG
  (3)  032334  104410                           TRAP    C$ESCAPE
  (3)  032336  0000 4                           .WORD   10000$-.
```

```
 7664  032340  005202                          INC     R2              ;NEXT DATA
 7665  032342  005205                          INC     R5              ;NEXT ADDRESS
 7666  032344  022705  000010                  CMP     #10,R5          ;DONE YET?
 7667  032350  001324                          BNE     1$              ;BR IF NO
 7668  032352                                  ENDSEG
  (3)  032352                          10000$:
  (3)  032352  104405                          TRAP    C$ESEG
 7669  032354                                  EXIT    TST
  (3)  032354  104432                          TRAP    C$EXIT
  (3)  032356  000012                          .WORD   L10155-.
 7670  032360     000      377      377  5$:    .BYTE   0,-1,-1,376,252,-1,-1,124
       032363     376      252      377
       032366     377      124
 7671
 7672                                 .EVEN
 7673  032370                         ENDTST
  (3)  032370                         L10155:
  (3)  032370  104401                          TRAP    C$ETST
 7674
 7675
 7676  032372                                  BADHEAD
  (2)                                         ;********************* TEST 68 *********************
 7677                                         ;*ALU TEST
 7678                                         ;*TEST OF ALU FUNCTION 2A W/C WITH C BIT SET
 7679                                         ;*ALU FUNCTION (A PLUS A PLUS C)        CODE-6
 7680                                         ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 7681                                         ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 7682  032372                                  BADHEAD
  (2)                                         ;********************* TEST 68 *********************
 7683
 7684  032372                         BGNTST
  (3)  032372                         T68::
 7685  032372                                  MYINT
  (1)  032372  013701  002716                 MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 7686  032376                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  032376  004537  003156                 JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 7687  032402  005005                          CLR     R5              ;MEM + SP ADDRESS
 7688  032404  012702  032564                 MOV     #5$,R2          ;POINTER TO CORRECT DATA
 7689  032410  004737  003640                 JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 7690  032414  002654                          MEMDAT                  ;POINTER TO DATA
 7691  032416  004737  004012                 JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 7692  032422  002664                          SPDAT                   ;POINTER TO DATA
 7693  032424                                  BGNSEG
  (3)  032424  104404                          TRAP    C$BSEG
 7694  032426  004737  004076          1$:    JSR     PC,SETC         ;SET C BIT!
 7695  032432  042737  000017  032450         BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7696  032440  050537  032450                 BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 7697  032444                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  032444  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7698  032450  010000                  2$:    010000                  ;LOAD MAR
 7699  032452  042737  000017  032470         BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 7700  032460  050537  032470                 BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 7701  032464                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=55304
  (1)  032464  004537  003244                 JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7702  032470  040540                  3$:    040400!<6*20>           ;BR   2A W/C
 7703  032472                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
  (1)  032472  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7704   032476  061224                        61224                   ;MOVE BR TO PORT4
7705   032500  111237  002636                MOVB    (R2),$GDDAT             ;PUT "WXPECTED" IN $GDDAT
7706   032504  116104  000004                MOVB    4(R1),R4        ;PUT "FOUND" IN R4
7707   032510  123704  002636                CMPB    $GDDAT,R4               ;DATA CORRECT?
7708   032514  001411                        BEQ     4$              ;BR IF YES
7709   032516                                ERROR   23,YES          ;ALU ERROR
  (5)  032530  104455                        TRAP    C$ERDF
  (6)  032532  000027                        .WORD   23
  (6)  032534  005226                        .WORD   EM23
  (6)  032536  007404                        .WORD   ERR23
7710   032540                        4$:     ESCAPE  SEG
  (3)  032540  104410                        TRAP    C$ESCAPE
  (3)  032542  000014                        .WORD   10000$-.
7711   032544  005202                        INC     R2              ;NEXT DATA
7712   032546  005205                        INC     R5              ;NEXT ADDRESS
7713   032550  022705  000010                CMP     #10,R5          ;DONE YET?
7714   032554  001324                        BNE     1$              ;BR IF NO
7715   032556                                ENDSEG
  (3)  032556                        10000$:
  (3)  032556  104405                        TRAP    C$ESEG
7716   032560                                EXIT    TST
  (3)  032560  104432                        TRAP    C$EXIT
  (3)  032562  000012                        .WORD   L10156-.
7717   032564     001     001     377  5$:     .BYTE   1,1,-1,-1,253,253,125,125
       032567     377     253     253
       032572     125     125
7718
7719                                  .EVEN
7720   032574                         ENDTST
  (3)  032574                         L10156:
  (3)  032574  104401                        TRAP    C$ETST
7721
7722
7723   032576                                BADHEAD
  (2)                                   ;********************* TEST 69 *********************
7724                                    ;*ALU TEST
7725                                    ;*TEST OF ALU FUNCTION SUB WITH C BIT SET
7726                                    ;*ALU FUNCTION (A-B)    CODE=16
7727                                    ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7728                                    ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7729   032576                           BADHEAD
  (2)                                   ;********************* TEST 69 *********************
7730
7731   032576                         BGNTST
  (3)  032576                         T69::
7732   032576                                MYINT
  (1)  032576  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7733   032602                                MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  032602  004537  003156                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
7734   032606  005005                        CLR     R5              ;MEM + SP ADDRESS
7735   032610  012702  032770                MOV #5$,R2              ;POINTER TO CORRECT DATA
7736   032614  004737  003640                JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7737   032620  002654                        MEMDAT                  ;POINTER TO DATA
7738   032622  004737  004012                JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7739   032626  002664                        SPDAT                   ;POINTER TO DATA
```

```
 7740  032630                                    BGNSEG
  (3)  032630  104404                           TRAP    C$BSEG
 7741  032632  004737  004076          1$:      JSR     PC,SETC         ;SET C BIT!
 7742  032636  042737  000017  032654          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7743  032644  050537  032654                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 7744  032650                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  032650  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7745  032654  010000                  2$:      010000                  ;LOAD MAR
 7746  032656  042737  000017  032674          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 7747  032664  050537  032674                  BIS     R5,3$           ;ADD ADDRESS  TO INSTRUCTION
 7748  032670                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  032670  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7749  032674  040740                  3$:      040400.<16*20>          ;BR   SUB
 7750  032676                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  032676  004537  003244                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 7751  032702  061224                          61224                   ;MOVE BR TO PORT4
 7752  032704  111237  002636                  MOVB    (R2),$GDDAT             ;PUT 'EXPECTED'' IN $GDDAT
 7753  032710  116104  000004                  MOVB    4(R1),R4        ;PUT 'FOUND'' IN R4
 7754  032714  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
 7755  032720  001411                          BEQ     4$              ;BR IF YES
 7756  032722                                    ERROR   23,YES          ;ALU ERROR
  (5)  032734  104455                          TRAP    C$ERDF
  (6)  032736  000027                          .WORD   23
  (6)  032740  005226                          .WORD   EM23
  (6)  032742  007404                          .WORD   ERR23
 7757  032744                          4$:      ESCAPE  SEG
  (3)  032744  104410                          TRAP    C$ESCAPE
  (3)  032746  000014                          .WORD   10000$-.
 7758  032750  005202                          INC     R2              ;NEXT DATA
 7759  032752  005205                          INC     R5              ;NEXT ADDRESS
 7760  032754  022705  000010                  CMP     #10,R5          ;DONE YET?
 7761  032760  001324                          BNE     1$              ;BR IF NO
 7762  032762                                    ENDSEG
  (3)  032762                          10000$:
  (3)  032762  104405                          TRAP    C$ESEG
 7763  032764                                    EXIT    TST
  (3)  032764  104432                          TRAP    C$EXIT
  (3)  032766  000012                          .WORD   L10157-.
 7764  032770     000     001     377  5$:      .BYTE   0,1,-1,0,0,253,125,0
       032773     000     000     253
       032776     125     000
 7765
 7766                                          .EVEN
 7767  033000                                  ENDTST
  (3)  033000                                  L10157:
  (3)  033000  104401                          TRAP    C$ETST
 7768
 7769
 7770  033002                                    BADHEAD
  (2)                                          ;********************* TEST 70 *********************
 7771                                          ;*ALU TEST
 7772                                          ;*TEST OF ALU FUNCTION ADD W/C WITH C BIT SET
 7773                                          ;*ALU FUNCTION (A PLUS B PLUS C)          CODE=01
 7774                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 7775                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 7776  033002                                    BADHEAD
```

```
 (2)                                        ;****************** TEST 70 ******************
7777
7778  033002                               BGNTST
 (3)  033002                               T70::
7779  033002                                     MYINT
 (1)  033002  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7780  033006                                     MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  033006  004537  003156                JSR     R5,..MSTCLR             ;CLEAR M8200,4,7
7781  033012  005005                        CLR     R5              ;MEM +SP ADDRESS
7782  033014  012702  033174                MOV     #5$,R2          ;POINTER TO CORRECT DATA
7783  033020  004737  003640                JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7784  033024  002654                        MEMDAT                  ;POINTER TO DATA
7785  033026  004737  004012                JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7786  033032  002664                        SPDAT                   ;POINTER TO DATA
7787  033034                                     BGNSEG
 (3)  033034  104404                        TRAP    C$BSEG
7788  033036  004737  004076        1$:     JSR     PC,SETC         ;SET C BIT!
7789  033042  042737  000017  033060       BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7790  033050  050537  033060                BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
7791  033054                                     ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
 (1)  033054  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7792  033060  010000                2$:     010000                  ;LOAD MAR
7793  033062  042737  000017  033100       BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7794  033070  050537  033100                BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
7795  033074                                     ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
 (1)  033074  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7796  033100  040420                3$:     040400.<01*20>          ;BR - ADD W/C
7797  033102                                     ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
 (1)  033102  004537  003244                JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7798  033106  061224                        61224                   ;MOVE BR TO PORT4
7799  033110  111237  002636                MOVB    (R2),$GDDAT             ;PUT 'EXPECTED'' IN $GDDAT
7800  033114  116104  000004                MOVB    4(R1),R4        ;PUT 'FOUND'' IN R4
7801  033120  123704  002636                CMPB    $GDDAT,R4               ;DATA CORRECT?
7802  033124  001411                        BEQ     4$              ;BR IF YES
7803  033126                                     ERROR   23,YES          ;ALU ERROR
 (5)  033140  104455                        TRAP    C$ERDF
 (6)  033142  000027                        .WORD   23
 (6)  033144  005226                        .WORD   EM23
 (6)  033146  007404                        .WORD   ERR23
7804  033150                         4$:     ESCAPE  SEG
 (3)  033150  104410                        TRAP    C$ESCAPE
 (3)  033152  000014                        .WORD   10000$-.
7805  033154  005202                        INC     R2              ;NEXT DATA
7806  033156  005205                        INC     R5              ;NEXT ADDRESS
7807  033160  022705  000010                CMP     #10,R5          ;DONE YET?
7808  033164  001324                        BNE     1$              ;BR IF NO
7809  033166                                     ENDSEG
 (3)  033166                        10000$:
 (3)  033166  104405                        TRAP    C$ESEG
7810  033170                                     EXIT    TST
 (3)  033170  104432                        TRAP    C$EXIT
 (3)  033172  000012                        .WORD   L10160-.
7811  033174     001     000     000  5$:     .BYTE   1,0,0,-1,253,0,0,125
      033177     377     253     000
      033202     000     125
7812
```

```
                                        .EVEN
                                        ENDTST
                                        L10160:
                                                TRAP    C$ETST

                                                BADHEAD
                                                ;********************* TEST 71 *********************
                                                ;*ALU TEST
                                                ;*TEST OF ALU FUNCTION SUB W/C WITH C BIT SET
                                                ;*ALU FUNCTION (A-B-C)  CODE=2
                                                ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
                                                BADHEAD
                                                ;********************* TEST 71 *********************

                                        BGNTST
                                        T71::
                                                MYINT
                                                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
                                                MSTCLR                  ;MASTER CLEAR M8200,4,7
                                                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
                                                CLR     R5              ;MEM + SP ADDRESS
                                                MOV     #5$,R2          ;POINTER TO CORRECT DATA
                                                JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
                                                MEMDAT                  ;POINTER TO DATA
                                                JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
                                                SPDAT                   ;POINTER TO DATA
                                                BGNSEG
                                                TRAP    C$BSEG
                                        1$:     JSR     PC,SETC         ;SET C BIT!
                                                BIC     #17,2$          ;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 INSTRUCITON, ROMCLK PC 55304
                                                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
                                        3$:     040400!<2*20>           ;BR   SUB W/C
                                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
                                                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
                                                61224                   ;MOVE BR TO PORT4
                                                MOVB    (R2),$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
```

```
 (3)  033356  000014                              .WORD   10000$-.
7853  033360  005202                              INC     R2              ;NEXT DATA
7854  033362  005205                              INC     R5              ;NEXT ADDRESS
7855  033364  022705  000010                      CMP     #10,R5                  ;DONE YET?
7856  033370  001324                              BNE     1$              ;BR IF NO
7857  033372                                      ENDSEG
 (3)  033372                              10000$:
 (3)  033372  104405                              TRAP    C$ESEG
7858  033374                                      EXIT    TST
 (3)  033374  104432                              TRAP    C$EXIT
 (3)  033376  000012                              .WORD   L10161-.
7859  033400     000     001     377  5$:         .BYTE   0,1,-1,0,0,253,125,0
      033403     000     000     253
      033406     125     000
7860
7861                                     .EVEN
7862  033410                             ENDTST
 (3)  033410                             L10161:
 (3)  033410  104401                              TRAP    C$ETST
7863
7864
7865  033412                                      BADHEAD
 (2)                                              ;****************** TEST 72 ******************
7866                                              ;*ALU TEST
7867                                              ;*TEST OF ALU FUNCTION INC A WITH C  BIT SET
7868                                              ;*ALU FUNCTION (A PLUS 1)       CODE=3
7869                                              ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7870                                              ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7871  033412                                      BADHEAD
 (2)                                              ;****************** TEST 72 ******************
7872
7873  033412                             BGNTST
 (3)  033412                             T72::
7874  033412                                      MYINT
 (1)  033412  013701  002716                      MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
7875  033416                                      MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1)  033416  004537  003156                      JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
7876  033422  005005                              CLR     R5              ;MEM + SP ADDRESS
7877  033424  012702  033604                      MOV     #5$,R2          ;POINTER TO CORRECT DATA
7878  033430  004737  003640                      JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7879  033434  002654                              MEMDAT                  ;POINTER TO DATA
7880  033436  004737  004012                      JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7881  033442  002664                              SPDAT                   ;POINTER TO DATA
7882  033444                                      BGNSEG
 (3)  033444  104404                              TRAP    C$BSEG
7883  033446  004737  004076              1$:     JSR     PC,SETC         ;SET C BIT.
7884  033452  042737  000017  033470              BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7885  033460  050537  033470                      BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
7886  033464                                      ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  033464  004537  003244                      JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7887  033470  010000                      2$:     010000                  ;LOAD MAR
7888  033472  042737  000017  033510              BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7889  033500  050537  033510                      BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
7890  033504                                      ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1)  033504  004537  003244                      JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
7891  033510  040460                      3$:     040400.<3*20>           ;BR _ INC A
```

```
 7892  033512                                    ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
  (1)  033512  004537  003244                    JSR     R5,..ROMCLK                     ;CLOCK INSTRUCTION
 7893  033516  061224                            61224                           ;MOVE BR TO PORT4
 7894  033520  111237  002636                    MOVB    (R2),$GDDAT                     ;PUT 'EXPECTED' IN $GDDAT
 7895  033524  116104  000004                    MOVB    4(R1),R4                ;PUT 'FOUND IN R4
 7896  033530  123704  002636                    CMPB    $GDDAT,R4                       ;DATA CORRECT?
 7897  033534  001411                            BEQ     4$                      ;BR IF YES
 7898  033536                                    ERROR   23,YES                  ;ALU ERROR
  (5)  033550  104455                            TRAP    C$ERDF
  (6)  033552  000027                            .WORD   23
  (6)  033554  005226                            .WORD   EM23
  (6)  033556  007404                            .WORD   ERR23
 7899  033560                              4$:   ESCAPE  SEG
  (3)  033560  104410                            TRAP    C$ESCAPE
  (3)  033562  000014                            .WORD   10000$-.
 7900  033564  005202                            INC     R2                      ;NEXT DATA
 7901  033566  005205                            INC     R5                      ;NEXT ADDRESS
 7902  033570  022705  000010                    CMP     #10,R5                  ;DONE YET?
 7903  033574  001324                            BNE     1$                      ;BR IF NO
 7904  033576                                    ENDSEG
  (3)  033576                              10000$:
  (3)  033576  104405                            TRAP    C$ESEG
 7905  033600                                    EXIT    TST
  (3)  033600  104432                            TRAP    C$EXIT
  (3)  033602  000012                            .WORD   L10162-.
 7906  033604     001     001     000      5$:   .BYTE   1,1,0,0,126,126,253,253
       033607     000     126     126
       033612     253     253
 7907
 7908                                      .EVEN
 7909  033614                              ENDTST
  (3)  033614                              L10162:
  (3)  033614  104401                            TRAP    C$ETST
 7910
 7911
 7912  033616                                    BADHEAD
  (2)                                           ;********** TEST 73 **********
 7913                                           ;*ALU TEST
 7914                                           ;*TEST OF ALU FUNCTION 2A WITH C BIT SET
 7915                                           ;*ALU FUNCTION (A PLUS A)       CODE=5
 7916                                           ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 7917                                           ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 7918  033616                                    BADHEAD
  (2)                                           ;********** TEST 73 **********
 7919
 7920  033616                              BGNTST
  (3)  033616                              T73::
 7921  033616                                    MYINT
  (1)  033616  013701  002716                    MOV     KMCSR,R1                ;GET DEVICE ADDRESS.
 7922  033622                                    MSTCLR                          ;MASTER CLEAR M8200,4,7
  (1)  033622  004537  003156                    JSR     R5,..MSTCLR                     ;CLEAR M8200,4,7
 7923  033626  005005                            CLR     R5                      ;MEM + SP ADDRESS
 7924  033630  012702  034010                    MOV     #5$,R2                  ;POINTER TO CORRECT DATA
 7925  033634  004737  003640                    JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
 7926  033640  002654                            MEMDAT                          ;POINTER TO DATA
 7927  033642  004737  004012                    JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
```

```
7928  033646  002664                          SPDAT                   ;POINTER TO DATA
7929  033650                                  BGNSEG
 (3)  033650  104404                          TRAP    C$BSEG
7930  033652  004737  004076          1$:     JSR     PC,SETC         ;SET C BIT:
7931  033656  042737  000017  033674          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7932  033664  050537  033674                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
7933  033670                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
 (1)  033670  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7934  033674  010000                  2$:     010000                  ;LOAD MAR
7935  033676  042737  000017  033714          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7936  033704  050537  033714                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
7937  033710                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
 (1)  033710  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7938  033714  040520                  3$:     040400.<5*20>           ;BR    2A
7939  033716                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
 (1)  033716  004537  003244                  JSR     R5,..ROMCLK             ;CLOCK INSTRUCTION
7940  033722  061224                          61224                   ;MOVE BR TO PORT4
7941  033724  111237  002636                  MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
7942  033730  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND IN R4
7943  033734  123704  002636                  CMPB    $GDDAT,R4               ;DATA CORRECT?
7944  033740  001411                          BEQ     4$              ;BR IF YES
7945  033742                                  ERROR   23,YES          ;ALU ERROR
 (5)  033754  104455                          TRAP    C$ERDF
 (6)  033756  000027                          .WORD   23
 (6)  033760  005226                          .WORD   EM23
 (6)  033762  007404                          .WORD   ERR23
7946  033764                          4$:     ESCAPE  SEG
 (3)  033764  104410                          TRAP    C$ESCAPE
 (3)  033766  000014                          .WORD   10000$-.
7947  033770  005202                          INC     R2                      ;NEXT DATA
7948  033772  005205                          INC     R5                      ;NEXT ADDRESS
7949  033774  022705  000010                  CMP     #10,R5                  ;DONE YET?
7950  034000  001324                          BNE     1$                      ;BR IF NO
7951  034002                                  ENDSEG
 (3)  034002                          10000$:
 (3)  034002  104405                          TRAP    C$ESEG
7952  034004                                  EXIT    TST
 (3)  034004  104432                          TRAP    C$EXIT
 (3)  034006  000012                          .WORD   L10163-.
7953  034010     000      000      376  5$:     .BYTE   0,0,376,376,252,252,124,124
      034013     376      252      252
      034016     124      124
7954
7955                                  .EVEN
7956  034020                          ENDTST
 (3)  034020                          L10163:
 (3)  034020  104401                          TRAP    C$ETST
7957
7958
7959  034022                                  BADHEAD
 (2)                                          ;********** TEST 74 **********
7960                                          ;*ALU TEST
7961                                          ;*TEST OF ALU FUNCTION A PLUS C WITH C BIT SET
7962                                          ;*ALU FUNCTION (A PLUS C)       CODE=4
7963                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7964                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
```

```
                                        BADHEAD
                                        ;******************** TEST 74 ********************

                                BGNTST
                                T74::
                                        MYINT
 (1) 034022  013701  002716             MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
 (1) 034026  004537  003156             JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
                                        CLR     R5              ;MEM + SP ADDRESS
                                        MOV     #5$,R2          ;POINTER TO CORRECT DATA
                                        JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
                                        MEMDAT                  ;POINTER TO DATA
                                        JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
                                        SPDAT                   ;POINTER TO DATA
                                        BGNSEG
 (3) 034054  104404                     TRAP    C$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
 (1) 034074  004537  003244             JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
                                2$:     010000                  ;LOAD MAR
                                        BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
                                        BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1) 034114  004537  003244             JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
                                3$:     040400!<4*20>           ;BR   A PLUS C
                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 (1) 034122  004537  003244             JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
                                        61224                   ;MOVE BR TO PORT4
                                        MOVB    (R2),$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
 (5) 034160  104455                     TRAP    C$ERDF
 (6) 034162  000027                     .WORD   23
 (6) 034164  005226                     .WORD   EM23
 (6) 034166  007404                     .WORD   ERR23
                                4$:     ESCAPE  SEG
 (3) 034170  104410                     TRAP    C$ESCAPE
 (3) 034172  000014                     .WORD   10000$-.
                                        INC     R2              ;NEXT DATA
                                        INC     R5              ;NEXT ADDRESS
                                        CMP     #10,R5          ;DONE YET?
                                        BNE     1$              ;BR IF NO
                                        ENDSEG
                                10000$:
 (3) 034206  104405                     TRAP    C$ESEG
                                        EXIT    TST
 (3) 034210  104432                     TRAP    C$EXIT
 (3) 034212  000012                     .WORD   L10164-.
                                5$:     .BYTE   1,1,0,0,126,126,253,253
```

```
8001
8002                                    .EVEN
8003  034224                            ENDTST
 (3)  034224                            L10164:
 (3)  034224  104401                            TRAP    C$ETST
8004
8005  034226                                    BADHEAD
 (2)                                            ;********************* TEST 75 *********************
8006                                            ;*ALU TEST
8007                                            ;*TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT SET
8008                                            ;*ALU FUNCTION (A-B-1)  CODE=17
8009                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8010                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
8011  034226                                    BADHEAD
 (2)                                            ;********************* TEST 75 ******* ************
8012
8013  034226                            BGNTST
 (3)  034226                            T75::
```

```
 8015  034226                                          MYINT
  (1)  034226  013701  002716                         MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 8016  034232                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  034232  004537  003156                         JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 8017  034236  005005                                  CLR     R5              ;MEM + SP ADDRESS
 8018  034240  012702  034420                         MOV     #5$,R2          ;POINTER TO CORRECT DATA
 8019  034244  004737  003640                         JSR     PC,MEMLD        ;LOAD 3 WORDS OF MAIN MEMORY
 8020  034250  002654                                  MEMDAT                  ;POINTER TO DATA
 8021  034252  004737  004012                         JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 8022  034256  002664                                  SPDAT                   ;POINTER TO DATA
 8023  034260                                          BGNSEG
  (3)  034260  104404                                 TRAP    C$BSEG
 8024  034262  004737  004076              1$:         JSR     PC,SETC         ;SET C BIT!
 8025  034266  042737  000017  034304                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 8026  034274  050537  034304                         BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 8027  034300                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
  (1)  034300  004537  003244                         JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 8028  034304  010000                        2$:      010000                  ;LOAD MAR
 8029  034306  042737  000017  034324                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 8030  034314  050537  034324                         BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 8031  034320                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
  (1)  034320  004537  003244                         JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 8032  034324  040760                        3$:      040400!<17*20>          ;BR   2'S COMP SUB
 8033  034326                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  034326  004537  003244                         JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 8034  034332  061224                                  61224                   ;MOVE BR TO PORT4
 8035  034334  111237  002636                         MOVB    (R2),$GDDAT             ;PUT 'EXPECTED'' IN $GDDAT
 8036  034340  116104  000004                         MOVB    4(R1),R4        ;PUT 'FOUND IN R4
 8037  034344  123704  002636                         CMPB    $GDDAT,R4               ;DATA CORRECT?
 8038  034350  001411                                  BEQ     4$              ;BR IF YES
 8039  034352                                          ERROR   23,YES          ;ALU ERROR
  (5)  034364  104455                                 TRAP    C$ERDF
  (6)  034366  000027                                 .WORD   23
  (6)  034370  005226                                 .WORD   EM23
  (6)  034372  007404                                 .WORD   ERR23
 8040  034374                              4$:         ESCAPE  SEG
  (3)  034374  104410                                 TRAP    C$ESCAPE
  (3)  034376  000014                                 .WORD   10000$-.
 8041  034400  005202                                  INC     R2              ;NEXT DATA
 8042  034402  005205                                  INC     R5              ;NEXT ADDRESS
 8043  034404  022705  000010                         CMP     #10,R5          ;DONE YET?
 8044  034410  001324                                  BNE     1$              ;BR IF NO
 8045  034412                                          ENDSEG
  (3)  034412                              10000$:
  (3)  034412  104405                                 TRAP    C$ESEG
 8046  034414                                          EXIT    TST
  (3)  034414  104432                                 TRAP    C$EXIT
  (3)  034416  000012                                 .WORD   L10165-.
 8047  034420     377      000      376   5$:         .BYTE   -1,0,376,-1,-1,252,124,-1
       034423     377      377      252
       034426     124      377
 8048
 8049                                             .EVEN
 8050  034430                                     ENDTST
  (3)  034430                                  .L10165:
  (3)  034430  104401                                 TRAP    C$ETST
```

```
 8051
 8052
 8053  034432                                  BADHEAD
  (2)                                          ;****************** TEST 76 ******************
 8054                                          ;*ALU TEST
 8055                                          ;*TEST OF ALU FUNCTION DEC A WITH C BIT SET
 8056                                          ;*ALU FUNCTION (A-1)    CODE=7
 8057                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 8058                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 8059  034432                                  BADHEAD
  (2)                                          ;****************** TEST 76 ******************
 8060
 8061  034432                        BGNTST
  (3)  034432                        T76::
 8062  034432                                  MYINT
  (1)  034432  013701  002716                MOV     KMCSR,R1        ;GET DEVICE ADDRESS.
 8063  034436                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
  (1)  034436  004537  003156                JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
 8064  034442  005005                        CLR     R5              ;MEM + SP ADDRESS
 8065  034444  012702  034624                MOV     #5$,R2          ;POINTER TO CORRECT DATA
 8066  034450  004737  003640                JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 8067  034454  002654                        MEMDAT                  ;POINTER TO DATA
 8068  034456  004737  004012                JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 8069  034462  002664                        SPDAT                   ;POINTER TO DATA
 8070  034464                                  BGNSEG
  (3)  034464  104404                        TRAP    C$BSEG
 8071  034466  004737  004076        1$:     JSR     PC,SETC         ;SET C BIT!
 8072  034472  042737  000017  034510        BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 8073  034500  050537  034510                BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 8074  034504                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
  (1)  034504  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 8075  034510  010000                2$:     010000                  ;LOAD MAR
 8076  034512  042737  000017  034530        BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 8077  034520  050537  034530                BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 8078  034524                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  034524  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 8079  034530  040560                3$:     040400!<7*20>           ;BR   DEC A
 8080  034532                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  (1)  034532  004537  003244                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
 8081  034536  061224                        61224                   ;MOVE BR TO PORT4
 8082  034540  111237  002636                MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
 8083  034544  116104  000004                MOVB    4(R1),R4        ;PUT "FOUND IN R4
 8084  034550  123704  002636                CMPB    $GDDAT,R4               ;DATA CORRECT?
 8085  034554  001411                        BEQ     4$              ;BR IF YES
 8086  034556                                  ERROR   23,YES          ;ALU ERROR
  (5)  034570  104455                        TRAP    C$ERDF
  (6)  034572  000027                        .WORD   23
  (6)  034574  005226                        .WORD   EM23
  (6)  034576  007404                        .WORD   ERR23
 8087  034600                        4$:     ESCAPE  SEG
  (3)  034600  104410                        TRAP    C$ESCAPE
  (3)  034602  000014                        .WORD   10000$-.
 8088  034604  005202                        INC     R2              ;NEXT DATA
 8089  034606  005205                        INC     R5              ;NEXT ADDRESS
 8090  034610  022705  000010                CMP     #10,R5          ;DONE YET?
 8091  034614  001324                        BNE     1$              ;BR IF NO
```

```
8092  034616                                        ENDSEG
 (3)  034616                                10000$:
 (3)  034616  104405                               TRAP    C$ESEG
8093  034620                                        EXIT    TST
 (3)  034620  104432                               TRAP    C$EXIT
 (3)  034622  000012                               .WORD   L10166-.
8094  034624     377     377     376    5$:       .BYTE   -1,-1,376,376,124,124,251,251
      034627     376     124     124
      034632     251     251
8095
8096                                       .EVEN
8097  034634                               ENDTST
 (3)  034634                               L10166:
 (3)  034634  104401                               TRAP    C$ETST
8098
8099
```

```
8101
8102
8103
8104
8105
8106
```

```
 8108                                   .SBTTL  HARDWARE PARAMETER CODING SECTION
 8109
 8110
 8111
 8112                                   ;////////////////////////////////////////////////////////////////////////
 8113                                   ;/ THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
 8114                                   ;/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
 8115                                   ;/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
 8116                                   ;/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
 8117                                   ;/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
 8118                                   ;/ WITH THE OPERATOR.
 8119                                   ;////////////////////////////////////////////////////////////////////////
 8120
 8121  034636                                   BGNHRD
  (3)  034636  000022                         .WORD L10167-L$HARD/2
  (3)  034640                          L$HARD::
 8122
 8123  034640                                   GPRMD   WMP,0,0,7,0,7,YES
  (4)  034640  000032                         .WORD   T$CODE
  (4)  034642  034704                         .WORD   WMP
  (4)  034644  000007                         .WORD   7
  (4)  034646  000000                         .WORD   T$LOLIM
  (4)  034650  000007                         .WORD   T$HILIM
 8124  034652                                   GPRMA   ADDRES,2,0,160000,177776,YES
  (4)  034652  001031                         .WORD   T$CODE
  (4)  034654  034763                         .WORD   ADDRES
  (4)  034656  160000                         .WORD   T$LOLIM
  (4)  034660  177776                         .WORD   T$HILIM
 8125  034662                                   GPRMA   VECTOR,4,0,0,770,YES
  (4)  034662  002031                         .WORD   T$CODE
  (4)  034664  035022                         .WORD   VECTOR
  (4)  034666  000000                         .WORD   T$LOLIM
  (4)  034670  000770                         .WORD   T$HILIM
 8126  034672                                   GPRMD   PRIRTY,6,0,7000,4,7,YES
  (4)  034672  003032                         .WORD   T$CODE
  (4)  034674  035064                         .WORD   PRIRTY
  (4)  034676  007000                         .WORD   7000
  (4)  034700  000004                         .WORD   T$LOLIM
  (4)  034702  000007                         .WORD   T$HILIM
 8127                                   ;        GPRMD   LNUNIT,10,0,3,0,3,YES
 8128                                   ;        GPRMD   SWPAC1,12,0,377,0,377,YES
 8129                                   ;        GPRMD   SWPAC2,14,0,377,0,377,YES
 8130                                   ;        GPRMD   LOOPBK,16,0,40000,0,1,YES
 8131
 8132  034704                                   ENDHRD
  (2)                                         .EVEN
  (3)  034704                          L10167:
 8133
 8134  034704  044127  041511  020110  WMP:     .ASCIZ  'WHICH MICRO-PROCESSOR:''
       034712  044515  051103  026517
       034720  051120  041517  051505
       034726  047523  035122     000
 8135  034733     060  046475  031070           .ASCIZ  '0-M8200,4-M8204,7-M8207''
       034740  030060  032054  046475
       034746  031070  032060  033454
       034754  046475  031070  033460
```

```
                034762      000
 8136           034763      115  041511  047522  ADDRES: .ASCIZ  /MICRO-PROCESSOR CSR ADDRESS : /
                034770   050055  047522  042503
                034776   051523  051117  041440
                035004   051123  040440  042104
                035012   042522  051523  035040
                035020   000040
 8137           035022   044515  051103  026517  VECTOR: .ASCIZ  /MICRO-PROCESSOR VECTOR ADDRESS : /
                035030   051120  041517  051505
                035036   047523  020122  042526
                035044   052103  051117  040440
                035052   042104  042522  051523
                035060   035040  000040
 8138           035064   044515  051103  026517  PRIRTY: .ASCIZ  /MICRO-PROCESSOR PRIORITY LEVEL : /
                035072   051120  041517  051505
                035100   047523  020122  051120
                035106   047511  044522  054524
                035114   046040  053105  046105
                035122   035040  000040
 8139           035126   044127  041511  020110  LNUNIT: .ASCIZ  /WHICH LINE UNIT (0-3)? 0=NONE,1=M8201,2=M8202,3=M8203 : /
                035134   044514  042516  052440
                035142   044516  020124  030050
                035150   031455  037451  030040
                035156   047075  047117  026105
                035164   036461  034115  030062
                035172   026061  036462  034115
                035200   030062  026062  036463
                035206   034115  030062  020063
                035214   020072      000
 8140           035217      123  044527  041524  SWPAC1: .ASCIZ  /SWITCH PACK #1 (DDCMP LINE #) : /
                035224   020110  040520  045503
                035232   021440  020061  042050
                035240   041504  050115  046040
                035246   047111  020105  024443
                035254   035040  000040
 8141           035260   053523  052111  044103  SWPAC2: .ASCIZ  /SWITCH PACK #2 (BM873 BOOT ADR) : /
                035266   050040  041501  020113
                035274   031043  024040  046502
                035302   033470  020063  047502
                035310   052117  040440  051104
                035316   020051  020072      000
 8142           035323      127  046111  020114  LOOPBK: .ASCIZ  /WILL TEST CONNECTOR(S) BE USED ? 0=NO,1-YES : /
                035330   042524  052123  041440
                035336   047117  042516  052103
                035344   051117  051450  020051
                035352   042502  052440  042523
                035360   020104  020077  036460
                035366   047516  030454  054475
                035374   051505  035040  000040
 8143
 8144                                    .EVEN
 8145
 8146
 8147
 8148
 8149
```

```
8150
```

```
                                .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
(3)  035402  000000                     .WORD L10170-L$SOFT/2
(3)  035404                     L$SOFT::

035404                                  ENDSFT
(2)                                     .EVEN
(3)  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
(2)                                     .EVEN
(4)  040000  000000                     .WORD   0
(4)  040002  000000                     .WORD   0
(3)  040004                     L$LAST::
        000114                          LTN.ED T$TESTNUM

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

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



        000001                  .END
```

```
ADDRES  034763           8124    8135#
ADR   = 000020 G         3852#
ASSEMB= 000010           3706
BIT0  = 000001 G         3852#   4123    4124    4833    6521
BIT00 = 000000 G         3852#
BIT01 = 000002 G         3852#
BIT02 = 000004 G         3852#
BIT03 = 000010 G         3852#
BIT04 = 000020 G         3852#
BIT05 = 000040 G         3852#
BIT06 = 000100 G         3852#
BIT07 = 000200 G         3852#
BIT08 = 000400 G         3852#
BIT09 = 001000 G         3852#
BIT1  = 000002 G         3852#   4121    4123    4124
BIT10 = 002000 G         3852#   4197    4205    5113    6355    6409
BIT11 = 004000 G         3852#   4900    4909
BIT12 = 010000 G         3852#   4609    4928    4937
BIT13 = 020000 G         3852#   4200    4208    4620
BIT14 = 040000 G         3852#
BIT15 = 100000 G         3852#   4624
BIT2  = 000004 G         3852#   4124
BIT3  = 000010 G         3852#
BIT4  = 000020 G         3852#
BIT5  = 000040 G         3852#
BIT6  = 000100 G         3852#   4106    4107
BIT7  = 000200 G         3852#   4107
BIT8  = 000400 G         3852#   4179    5150    6357    6411
BIT9  = 001000 G         3852#   4871    4880    5113    5148    5150    6355    6409
BOE   = 000400 G         3852#
CLKX    002570           3909#
CLRALL  003274           4127#
CLRC    004060           4310#   6615    6661    6707    6753    6799    6845    6891    6937    6983    7031    7079    7127
                         7174    7221    7268    7316
CORMAX  035404           6517    6549    8178#
CRAMA   003522           4206    4215#
CZDMP   002000 G         3712#
C$AU  = 000052           3706#   4719
C$AUTO= 000061           3706#   4673
C$BRK = 000022           3706#
C$BSEG- 000004           3706#   4795    4810    4834    4869    4879    4898    4908    4926    4936    4955    4993    5006
                         5033    5046    5073    5088    5112    5123    5174    5192    5223    5241    5272    5290    5321
                         5339    5370    5393    5429    5453    5490    5508    5539    5557    5588    5606    5637    5655
                         5686    5704    5735    5753    5784    5802    5833    5851    5882    5909    5939    5957    5992
                         6015    6058    6086    6575    6614    6660    6706    6752    6798    6844    6890    6936    6982
                         7030    7078    7126    7173    7220    7267    7315    7362    7411    7458    7505    7552    7599
                         7646    7693    7740    7787    7835    7882    7929    7976    8023    8070
C$BSUB= 000002           3706#   5991
C$CEFG- 000045           3706#
C$CLCK= 000062           3706#
C$CLEA= 000012           3706#   4686
C$CLOS- 000035           3706#
C$CLP1= 000006           3706#
C$CVEC- 000036           3706#
C$DCLN= 000044           3706#
C$DODU- 000051           3706#   4669
```

```
C$DRPT= 000024   3706#
C$DU  = 000053   3706#  4703
C$EDIT= 000003   3706#  3747
C$ERDF= 000055   3706#  4282  4751  4768  4801  4815  4841  4847  4876  4884  4905  4913  4933
                 4941  4963  4969  4999  5013  5039  5054  5079  5093  5119  5128  5156  5184
                 5203  5233  5252  5282  5301  5331  5350  5384  5408  5443  5469  5500  5519
                 5549  5568  5598  5617  5647  5666  5696  5715  5745  5764  5794  5813  5843
                 5862  5900  5918  5949  5969  6008  6033  6077  6098  6126  6128  6151  6153
                 6190  6223  5227  6261  6295  6328  6381  6422  6459  6469  6502  6538  6589
                 6630  6676  6722  6768  6814  6860  6906  6952  6998  7046  7094  7142  7189
                 7236  7283  7331  7378  7427  7474  7521  7568  7615  7662  7709  7756  7803
                 7851  7898  7945  7992  8039  8086
C$ERHR= 000056   3706#
C$ERRO= 000060   3706#
C$ERSF= 000054   3706#
C$ERSO= 000057   3706#
C$ESCA= 000010   3706#  4744  4755  4802  4816  4842  4848  4877  4885  4906  4934  4964  4970
                 5000  5014  5040  5055  5080  5094  5120  5129  5158  5185  5204  5234  5253
                 5283  5302  5332  5351  5385  5409  5445  5470  5501  5520  5550  5569  5599
                 5618  5648  5667  5697  5716  5746  5765  5795  5814  5844  5863  5901  5919
                 5950  5970  6009  6034  6078  6099  6262  6296  6329  6539  6590  6631  6677
                 6723  6769  6815  6861  6907  6953  6999  7047  7095  7143  7190  7237  7284
                 7332  7379  7428  7475  7522  7569  7616  7663  7710  7757  7804  7852  7899
                 7946  7993  8040  8087
C$ESEG= 000005   3706#  4807  4821  4851  4878  4886  4907  4914  4935  4942  4973  5004  5019
                 5044  5060  5087  5099  5121  5130  5189  5209  5238  5258  5287  5307  5336
                 5356  5390  5415  5450  5476  5505  5525  5554  5574  5603  5623  5652  5672
                 5701  5721  5750  5770  5799  5819  5848  5868  5906  5925  5954  5977  6013
                 6039  6083  6104  6591  6636  6682  6728  6774  6820  6866  6912  6958  7004
                 7052  7100  7148  7195  7242  7289  7337  7384  7433  7480  7527  7574  7621
                 7668  7715  7762  7809  7857  7904  7951  7998  8045  8092
C$ESUB= 000003   3706#  6044
C$ETST- 000001   3706#  4756  4770  4822  4852  4888  4916  4944  4974  5020  5061  5100  5131
                 5160  5210  5259  5308  5357  5416  5477  5526  5575  5624  5673  5722  5771
                 5820  5869  5926  5978  6045  6105  6131  6156  6193  6231  6266  6300  6353
                 6383  6427  6472  6504  6557  6592  6641  6687  6733  6779  6825  6871  6917
                 6963  7010  7057  7106  7153  7200  7247  7294  7342  7389  7438  7485  7532
                 7579  7626  7673  7720  7767  7814  7862  7909  7956  8003  8050  8097
C$EXIT= 000032   3706#  6263  6297  6330  6553  6637  6683  6729  6775  6821  6867  6913  6959
                 7005  7053  7101  7149  7196  7243  7290  7338  7385  7434  7481  7528  7575
                 7622  7669  7716  7763  7810  7858  7905  7952  7999  8046  8093
C$GETB= 000026   3706#
C$GETW= 000027   3706#
C$GMAN- 000043   3706#
C$GPHR= 000042   3706#  4574
C$GPLO= 000030   3706#
C$GPRI= 000040   3706#
C$INIT- 000011   3706#  4652
C$INLP= 000020   3706#
C$MANI= 000050   3706#
C$MEM = 000031   3706#
C$MSG = 000023   3706#  4455  4456  4457  4458  4459  4460  4461  4462  4463  4464  4465  4467
                 4469  4470  4471  4472  4473  4474  4475  4476  4477  4478  4479  4480  4481
                 4482  4483  4484  4485  4486  4487  4488  4493
C$OPEN  000034   3706#
C$PNTB- 000014   3706#  4455  4456  4457  4458  4459  4460  4461  4462  4463  4464  4465  4467
```

```
                        4469    4470    4471    4472    4473    4474    4475    4476    4477    4478    4479    4480    4481
                        4482    4483    4484    4485    4486    4487    4488
C$PNTF= 000017           3706#   4491    4492
C$PNTS= 000016           3706#
C$PNTX= 000015           3706#
C$QIO = 000377           3706#
C$RDBU= 000007           3706#
C$REFG= 000047           3706#   4551    4554    4557    4561
C$RESE= 000033           3706#   4684    4702    6114    6189    6239
C$REVI= 000003           3706#   3747
C$RFLA= 000021           3706#
C$RPT = 000025           3706#   4523
C$SEFG= 000046           3706#
C$SPRI= 000041           3706#   6120    6124    6145    6149    6168    6183    6205    6221
C$SVEC= 000037           3706#
C$TPRI= 000013           3706#
DFPTBL  002364 G         3793#
DH0     005716           4398#   4465    4470    4471    4473    4474    4476    4483    4484
DH1     005717           4400#   4458    4459
DH2     005750           4401#   4455    4456    4457    4463    4464    4467    4472    4477    4478    4480    4482    4485
                         4486    4487    4488
DH27    006030           4404#   4479    4481
DH3     005764           4402#   4460    4461    4462
DH4     006014           4403#   4469    4475
DIAGMC= 000000           3706
EF.CON= 000036 G         3852#   4557
EF.NEW= 000035 G         3852#   4554
EF.PWR= 000034 G         3852#
EF.RES= 000037 G         3852#   4561
EF.STA= 000040 G         3852#   4551
EM1     004261           4354#   4491
EM10    004622           4366#
EM11    004654           4367#   6261    6295    6328    6381    6422    6538
EM12    004665           4368#
EM13    004707           4369#   6459    6469    6502
EM14    004731           4370#
EM15    004754           4371#   6630    6676    6722    6768    6814    6860    6906    6952    6998    7046    7094    7142
                         7189    7236    7283    7331    7474
EM16    005017           4373#
EM17    005064           4375#
EM2     004307           4355#   4815    4847    5079    5093
EM20    005113           4376#
EM21    005160           4377#
EM22    005207           4378#
EM23    005226           4379#   7378    7427    7521    7568    7615    7662    7709    7756    7803    7851    7898    7945
                         7992    8039    8086
EM24    005267           4380#
EM25    005453           4385#
EM26    005474           4386#   4768    4801    4876    4884    4905    4913    4933    4941    5119    5128
EM27    005525           4387#   4841    4963    4969    4999    5013    5039    5054    5184    5203    5233    5252    5282
                         5301    5331    5350    5384    5408    5443    5469
EM28    005554           4388#   5156
EM29    005605           4389#   5500    5519    5549    5568    5598    5617    5647    5666    5696    5715    5745    5764
                         5794    5813    5843    5862    5900
EM3     004416           4357#   5949    5969
EM30    004353           4356#   5918
```

```
EM31    005312          4381#  6126   6151   6223
EM32    005341          4382#  6128   6153   6227
EM33    005400          4383#  6190
EM34    005435          4384#  6589
EM35    005064          4374#
EM36    005640          4282   4391#
EM37    005715          4392#  4751
EM4     004444          4359#  6008   6033   6077
EM5     004472          4361#  6098
EM6     004533          4363#
EM7     004561          4364#
ENDBUG  003244          4115#
ENDIT   011254          4558   4642   4646   4649   4651#
ERRFLG  002560          3905#
ERR1    006054 G        4455#
ERR10   006576 G        4462#
ERR11   006660 G        4463#  6261   6295   6328   6381   6422   6538
ERR12   006736 G        4464#
ERR13   007014 G        4465#  6459   6469   6502
ERR14   007042 G        4467#
ERR15   007120 G        4469#  6630   6676   6722   6768   6814   6860   6906   6952   6998   7046   7094   7142
                        7089   7236   7283   7331   7474
ERR16   007176 G        4470#
ERR17   007224 G        4471#
ERR2    006132 G        4456#  4815   4847   5079   5093
ERR20   007252 G        4472#
ERR21   007330 G        4473#
ERR22   007356 G        4474#
ERR23   007404 G        4475#  7378   7427   7521   7568   7615   7662   7709   7756   7803   7851   7898   7945
                        7992   8039   8086
ERR24   007462 G        4476#
ERR25   007510 G        4477#
ERR26   007566 G        4478#  4768   4801   4876   4884   4905   4913   4933   4941   5119   5128
ERR27   007644 G        4479#  4841   4963   4969   4999   5013   5039   5054   5184   5203   5233   5252   5282
                        5301   5331   5350   5384   5408   5443   5469
ERR28   007726 G        4480#  5156
ERR29   010004 G        4481#  5500   5519   5549   5568   5598   5617   5647   5666   5696   5715   5745   5764
                        5794   5813   5843   5862   5900
ERR3    006210 G        4457#  5949   5969
ERR30   010066 G        4482#  5918
ERR31   010144 G        4483#  6126   6151   6223
ERR32   010172 G        4484#  6128   6153   6227
ERR33   010220 G        4485#  6190
ERR34   010276 G        4486#  6589
ERR35   010354 G        4487#
ERR36   010432 G        4282   4488#
ERR37   010510 G        4490#  4751
ERR4    006266 G        4458#  6008   6033   6077
ERR5    006350 G        4459#  6098
ERR6    006432 G        4460#
ERR7    006514 G        4461#
EVL     000004 G        3852#
E$END   002100          3706#
E$LOAD= 000035          3706#  3747
FLAG    002620          3922#
FM1     004114          4343#  4455   4456   4457   4458   4459   4460   4461   4462   4463   4464   4465   4467
```

```
                    4469    4470    4471    4472    4473    4474    4475    4476    4477    4478    4479    4480    4481
                    4482    4483    4484    4485    4486    4487    4488    4491
FTIME   002646      3933#   4543    4547*
F$AU    = 000015    3706#   4718    4719
F$AUTO= 000020      3706#   4655    4673
F$BGN   = 000040    3706#   3712    3756    4455    4456    4457    4458    4459    4460    4461    4462    4463    4464
                    4465    4467    4469    4470    4471    4472    4473    4474    4475    4476    4477    4478    4479
                    4480    4481    4482    4483    4484    4485    4486    4487    4488    4490    4506    4514    4537
                    4655    4683    4700    4718    4737    4744    4755    4756    4762    4770    4790    4795    4802
                    4810    4816    4822    4830    4834    4842    4848    4852    4867    4869    4877    4879    4885
                    4888    4896    4898    4906    4908    4916    4924    4926    4934    4936    4944    4951    4955
                    4964    4970    4974    4988    4993    5000    5006    5014    5020    5028    5033    5040    5046
                    5055    5061    5069    5073    5080    5088    5094    5100    5108    5112    5120    5123    5129
                    5131    5144    5158    5160    5168    5174    5185    5192    5204    5210    5218    5223    5234
                    5241    5253    5259    5267    5272    5283    5290    5302    5308    5316    5321    5332    5339
                    5351    5357    5365    5370    5385    5393    5409    5416    5424    5429    5445    5453    5470
                    5477    5485    5490    5501    5508    5520    5526    5534    5539    5550    5557    5569    5575
                    5583    5588    5599    5606    5618    5624    5632    5637    5648    5655    5667    5673    5681
                    5686    5697    5704    5716    5722    5730    5735    5746    5753    5765    5771    5779    5784
                    5795    5802    5814    5820    5828    5833    5844    5851    5863    5869    5877    5882    5901
                    5909    5919    5926    5934    5939    5950    5957    5970    5978    5986    5991    5992    6009
                    6015    6034    6044    6045    6053    6058    6078    6086    6099    6105    6112    6131    6138
                    6156    6164    6193    6201    6231    6238    6262    6263    6266    6273    6296    6297    6300
                    6307    6329    6330    6333    6341    6383    6395    6427    6436    6472    6480    6504    6512
                    6539    6553    6557    6568    6575    6590    6592    6605    6614    6631    6637    6641    6651
                    6660    6677    6683    6687    6697    6706    6723    6729    6733    6743    6752    6769    6775
                    6779    6789    6798    6815    6821    6825    6835    6844    6861    6867    6871    6881    6890
                    6907    6913    6917    6927    6936    6953    6959    6963    6973    6982    6999    7005    7010
                    7021    7030    7047    7053    7057    7068    7078    7095    7101    7106    7117    7126    7143
                    7149    7153    7164    7173    7190    7196    7200    7211    7220    7237    7243    7247    7258
                    7267    7284    7290    7294    7306    7315    7332    7338    7342    7353    7362    7379    7385
                    7389    7402    7411    7428    7434    7438    7449    7458    7475    7481    7485    7496    7505
                    7522    7528    7532    7543    7552    7569    7575    7579    7590    7599    7616    7622    7626
                    7637    7646    7663    7669    7673    7684    7693    7710    7716    7720    7731    7740    7757
                    7763    7767    7778    7787    7804    7810    7814    7826    7835    7852    7858    7862    7873
                    7882    7899    7905    7909    7920    7929    7946    7952    7956    7967    7976    7993    7999
                    8003    8013    8023    8040    8046    8050    8061    8070    8087    8093    8097    8121    8164
                    8175
F$CLEA= 000007      3706#   4683    4686
F$DU    = 000016    3706#   4700    4703
F$END   = 000041    3706#   3712    4455    4456    4457    4458    4459    4460    4461    4462    4463    4464    4465
                    4467    4469    4470    4471    4472    4473    4474    4475    4476    4477    4478    4479    4480
                    4481    4482    4483    4484    4485    4486    4487    4488    4493    4514    4523    4652    4673
                    4686    4703    4719    4737    4744    4755    4756    4762    4770    4790    4802    4807    4816
                    4821    4822    4830    4842    4848    4851    4852    4867    4877    4878    4885    4886    4888
                    4896    4906    4907    4914    4916    4924    4934    4935    4942    4944    4951    4964    4970
                    4973    4974    4988    5000    5004    5014    5019    5020    5028    5040    5044    5055    5060
                    5061    5069    5080    5087    5094    5099    5100    5108    5120    5121    5129    5130    5131
                    5144    5158    5160    5168    5185    5189    5204    5209    5210    5218    5234    5238    5253
                    5258    5259    5267    5283    5287    5302    5307    5308    5316    5332    5336    5351    5356
                    5357    5365    5385    5390    5409    5415    5416    5424    5445    5450    5470    5476    5477
                    5485    5501    5505    5520    5525    5526    5534    5550    5554    5569    5574    5575    5583
                    5599    5603    5618    5623    5624    5632    5648    5652    5667    5672    5673    5681    5697
                    5701    5716    5721    5722    5730    5746    5750    5765    5770    5771    5779    5795    5799
                    5814    5819    5820    5828    5844    5848    5863    5868    5869    5877    5901    5906    5919
                    5925    5926    5934    5950    5954    5970    5977    5978    5986    5991    6009    6013    6034
```

```
                        6039  6044  6045  6053  6078  6083  6099  6104  6105  6112  6131  6138  6156
                        6164  6193  6201  6231  6238  6262  6263  6266  6273  6296  6297  6300  6307
                        6329  6330  6333  6341  6383  6395  6427  6436  6472  6480  6504  6512  6539
                        6553  6557  6568  6590  6591  6592  6605  6631  6636  6637  6641  6651  6677
                        6682  6683  6687  6697  6723  6728  6729  6733  6743  6769  6774  6775  6779
                        6789  6815  6820  6821  6825  6835  6861  6866  6867  6871  6881  6907  6912
                        6913  6917  6927  6953  6958  6959  6963  6973  6999  7004  7005  7010  7021
                        7047  7052  7053  7057  7068  7095  7100  7101  7106  7117  7143  7148  7149
                        7153  7164  7190  7195  7196  7200  7211  7237  7242  7243  7247  7258  7284
                        7289  7290  7294  7306  7332  7337  7338  7342  7353  7379  7384  7385  7389
                        7402  7428  7433  7434  7438  7449  7475  7480  7481  7485  7496  7522  7527
                        7528  7532  7543  7569  7574  7575  7579  7590  7616  7621  7622  7626  7637
                        7663  7668  7669  7673  7684  7710  7715  7716  7720  7731  7757  7762  7763
                        7767  7778  7804  7809  7810  7814  7826  7852  7857  7858  7862  7873  7899
                        7904  7905  7909  7920  7946  7951  7952  7956  7967  7993  7998  7999  8003
                        8013  8040  8045  8046  8050  8061  8087  8092  8093  8097  8132  8167  8175
F$HARD= 000004           3706#  8121  8132
F$HW  = 000013           3706#  3793  3808
F$INIT= 000006           3706#  4537  4652
F$JMP - 000050           3706#  4514  6263  6297  6330  6553  6637  6683  6729  6775  6821  6867  6913
                        6959  7005  7053  7101  7149  7196  7243  7290  7338  7385  7434  7481  7528
                        7575  7622  7669  7716  7763  7810  7858  7905  7952  7999  8046  8093
F$MOD - 000000           3706#  3712  8175
F$MSG = 000011           3706#  4455  4456  4457  4458  4459  4460  4461  4462  4463  4464  4465  4467
                        4469  4470  4471  4472  4473  4474  4475  4476  4477  4478  4479  4480  4481
                        4482  4483  4484  4485  4486  4487  4488  4490  4493
F$PROT= 000021           3706#  3756  3760
F$PWR = 000017           3706#
F$RPT = 000012           3706#  4506  4523
F$SEG = 000003           3706#  4795  4807  4810  4821  4834  4851  4869  4878  4879  4886  4898  4907
                        4908  4914  4926  4935  4936  4942  4955  4973  4993  5004  5006  5019  5033
                        5044  5046  5060  5073  5087  5088  5099  5112  5121  5123  5130  5174  5189
                        5192  5209  5223  5238  5241  5258  5272  5287  5290  5307  5321  5336  5339
                        5356  5370  5390  5393  5415  5429  5450  5453  5476  5490  5505  5508  5525
                        5539  5554  5557  5574  5588  5603  5606  5623  5637  5652  5655  5672  5686
                        5701  5704  5721  5735  5750  5753  5770  5784  5799  5802  5819  5833  5848
                        5851  5868  5882  5906  5909  5925  5939  5954  5957  5977  5992  6013  5015
                        6039  6058  6083  6086  6104  6575  6591  6614  6636  6660  6682  6706  6728
                        6752  6774  6798  6820  6844  6866  6890  6912  6936  6958  6982  7004  7030
                        7052  7078  7100  7126  7148  7173  7195  7220  7242  7267  7289  7315  7337
                        7362  7384  7411  7433  7458  7480  7505  7527  7552  7574  7599  7621  7646
                        7668  7693  7715  7740  7762  7787  7809  7835  7857  7882  7904  7929  7951
                        7976  7998  8023  8045  8070  8092
F$SOFT- 000005           3706#  8164  8167
F$SRV = 000010           3706#
F$SUB = 000002           3706#  5991  6044
F$SW  = 000014           3706#  3822  3825
F$TEST= 000001           3706#  4737  4756  4762  4770  4790  4822  4830  4852  4867  4888  4896  4916
                        4924  4944  4951  4974  4988  5020  5028  5061  5069  5100  5108  5131  5144
                        5160  5168  5210  5218  5259  5267  5308  5316  5357  5365  5416  5424  5477
                        5485  5526  5534  5575  5583  5624  5632  5673  5681  5722  5730  5771  5779
                        5820  5828  5869  5877  5926  5934  5978  5986  6045  6053  6105  6112  6131
                        6138  6156  6164  6193  6201  6231  6238  6266  6273  6300  6307  6333  6341
                        6383  6395  6427  6436  6472  6480  6504  6512  6557  6568  6592  6605  6641
                        6651  6687  6697  6733  6743  6779  6789  6825  6835  6871  6881  6917  6927
                        6963  6973  7010  7021  7057  7068  7106  7117  7153  7164  7200  7211  7247
```

```
                        7258    7294    7306    7342    7353    7389    7402    7438    7449    7485    7496    7532    7543
                        7579    7590    7626    7637    7673    7684    7720    7731    7767    7778    7814    7826    7862
                        7873    7909    7920    7956    7967    8003    8013    8050    8061    8097
GE'PRM  010712          4562    4570#   4575
G$CNTO= 000200          3706#
G$DELM= 000372          3706#
G$DISP= 000003          3706#
G$EXCP= 000400          3706#
G$HILI= 000002          3706#
G$LOLI= 000001          3706#
G$NO   = 000000         3706#
G$OFFS- 000400          3706#   8123    8124    8125    8126
G$OFSI- 000376          3706#   8123    8124    8125    8126
G$PRMA= 000001          3706#   8124    8125
G$PRMD= 000002          3706#   8123    8126
G$PRML= 000000          3706#
G$RADA= 000140          3706#
G$RADB= 000000          3706#
G$RADD= 000040          3706#
G$RADL= 000120          3706#
G$RADO= 000020          3706#   8123    8124    8125    8126
G$XFER= 000004          3706#
G$YES  = 000010         3706#   8123    8124    8125    8126
HELP   = 000000         3693#   3739    3750    3773    4022    4508    4516
HOE    = 100000 G       3852#
IBE    - 010000 G       3852#
IDU    - 000040 G       3852#
IER    - 020000 G       3852#
INIFLG  002674          3947#
INSTU   003536          4207    4217#
ISR    - 000100 G       3852#
IXE    = 004000 G       3852#
I$AU   - 000041         3706#   4718#   4719#
I$AUTO- 000041          3706#   4655#   4673#
I$CLN  = 000041         3706#   4683#   4686#
I$DU   = 000041         3706#   4700#   4703#
I$HRD  = 000041         8121#   8132#
I$INIT= 000041          3706#   4537#   4652#
I$MOD  = 000041         3706#   3712#   8175#
I$MSG  - 000041         3706#   4455#   4456#   4457#   4458#   4459#   4460#   4461#   4462#   4463#   4464#   4465#   4467#
                        4469#   4470#   4471#   4472#   4473#   4474#   4475#   4476#   4477#   4478#   4479#   4480#   4481#
                        4482#   4483#   4484#   4485#   4486#   4487#   4488#   4490#   4493#
I$PROT  000C40          3706#   3756#
I$PTAB- 000041          3706#
I$PWR  - 000041         3706#
I$RPT  - 000041         3706#   4506#   4523#
I$SEG  = 000041         3706#   4737    4762    4790    4795#   4802    4807#   4810#   4816    4821#   4830    4834#   4842
                        4848    4851#   4867    4869#   4877    4878#   4879#   4885    4886#   4896    4898#   4906    4907#
                        4908#   4914#   4924    4926#   4934    4935#   4936#   4942#   4951    4955#   4964    4970    4973#
                        4988    4993#   5000    5004#   5006#   5014    5019#   5028    5033#   5040    5044#   5046#   5055
                        5060#   5069    5073#   5080    5087#   5088#   5094    5099#   5108    5112#   5120    5121#   5123#
                        5129    5130#   5144    5168    5174#   5185    5189#   5192#   5204    5209#   5218    5223#   5234
                        5238#   5241#   5253    5258#   5267    5272#   5283    5287#   5290#   5302    5307#   5316    5321#
                        5332    5336#   5339#   5351    5356#   5365    5370#   5385    5390#   5393#   5409    5415#   5424
                        5429#   5445    5450#   5453#   5470    5476#   5485    5490#   5501    5505#   5508#   5520    5525#
                        5534    5539#   5550    5554#   5557#   5569    5574#   5583    5588#   5599    5603#   5606#   5618
```

```
                    5623#  5632   5637#  5648   5652#  5655#  5667   5672#  5681   5686#  5697   5701#  5704#
                    5716   5721#  5730   5735#  5746   5750#  5753#  5765   5770#  5779   5784#  5795   5799#
                    5802#  5814   5819#  5828   5833#  5844   5848#  5851#  5863   5868#  5877   5882#  5901
                    5906#  5909#  5919   5925#  5934   5939#  5950   5954#  5957#  5970   5977#  5986   5991
                    5992#  6009   6013#  6015#  6039#  6053   6058#  6078   6083#  6086#  6099   6104#  6112
                    6138   6164   6201   6238   6273   6307   6341   6395   6436   6480   6512   6568   6575#
                    6590   6591#  6605   6614#  6631   6636#  6651   6660#  6677   6682#  6697   6706#  6723
                    6728#  6743   6752#  6769   6774#  6789   6798#  6815   6820#  6835   6844#  6861   6866#
                    6881   6890#  6907   6912#  6927   6936#  6953   6958#  6973   6982#  6999   7004#  7021
                    7030#  7047   7052#  7068   7078#  7095   7100#  7117   7126#  7143   7148#  7164   7173#
                    7190   7195#  7211   7220#  7237   7242#  7258   7267#  7284   7289#  7306   7315#  7332
                    7337#  7353   7362#  7379   7384#  7402   7411#  7428   7433#  7449   7458#  7475   7480#
                    7496   7505#  7522   7527#  7543   7552#  7569   7574#  7590   7599#  7616   7621#  7637
                    7646#  7663   7668#  7684   7693#  7710   7715#  7731   7740#  7757   7762#  7778   7787#
                    7804   7809#  7826   7835#  7852   7857#  7873   7882#  7899   7904#  7920   7929#  7946
                    7951#  7967   7976#  7993   7998#  8013   8023#  8040   8045#  8061   8070#  8087   8092#
I$SETU= 000041       3706#
I$SFT = 000041       8164#  8167#
I$SRV = 000041       3706#
I$SUB = 000041       3706#  4737   4762   4790   4830   4867   4896   4924   4951   4988   5028   5069   5108
                    5144   5168   5218   5267   5316   5365   5424   5485   5534   5583   5632   5681   5730
                    5779   5828   5877   5934   5986   5991#  6044#  6053   6112   6138   6164   6201   6238
                    6273   6307   6341   6395   6436   6480   6512   6568   6605   6651   6697   6743   6789
                    6835   6881   6927   6973   7021   7068   7117   7164   7211   7258   7306   7353   7402
                    7449   7496   7543   7590   7637   7684   7731   7778   7826   7873   7920   7967   8013
                    8061
I$TST = 000041       3706#  4737#  4744   4755   4756#  4762#  4770#  4790#  4822#  4830#  4852#  4867#  4888#
                    4896#  4916#  4924#  4944#  4951#  4974#  4988#  5020#  5028#  5061#  5069#  5100#  5108#
                    5131#  5144#  5158   5160#  5168#  5210#  5218#  5259#  5267#  5308#  5316#  5357#  5365#
                    5416#  5424#  5477#  5485#  5526#  5534#  5575#  5583#  5624#  5632#  5673#  5681#  5722#
                    5730#  5771#  5779#  5820#  5828#  5869#  5877#  5926#  5934#  5978#  5986#  5991   6034
                    6045#  6053#  6105#  6112#  6131#  6138#  6156#  6164#  6193#  6201#  6231#  6238#  6262
                    6263   6266#  6273#  6296   6297   6300#  6307#  6329   6330   6333#  6341#  6383#  6395#
                    6427#  6436#  6472#  6480#  6504#  6512#  6539   6553   6557#  6568#  6592#  6605#  6637
                    6641#  6651#  6683   6687#  6697#  6729   6733#  6743#  6775   6779#  6789#  6821   6825#
                    6835#  6867   6871#  6881#  6913   6917#  6927#  6959   6963#  6973#  7005   7010#  7021#
                    7053   7057#  7068#  7101   7106#  7117#  7149   7153#  7164#  7196   7200#  7211#  7243
                    7247#  7258#  7290   7294#  7306#  7338   7342#  7353#  7385   7389#  7402#  7434   7438#
                    7449#  7481   7485#  7496#  7528   7532#  7543#  7575   7579#  7590#  7622   7626#  7637#
                    7669   7673#  7684#  7716   7720#  7731#  7763   7767#  7778#  7810   7814#  7826#  7858
                    7862#  7873#  7905   7909#  7920#  7952   7956#  7967#  7999   8003#  8013#  8046   8050#
                    8061#  8093   8097#
J$JMP = 000167       3706#  4514
KMACTV  002610       3918#
KMCSR   002716       3983#  4578*  4657   4738   4763   4791   4808   4835   4870   4899   4927   5070   5085
                    5111   5145   5173   5222   5271   5320   5369   5428   5489   5538   5587   5636   5685
                    5734   5783   5832   5881   5938   5987   6057   6113   6139   6165   6202   6241   6274
                    6308   6343   6396   6437   6481   6513   6569   6606   6652   6698   6744   6790   6836
                    6882   6928   6974   7022   7069   7118   7165   7212   7259   7307   7354   7403   7450
                    7497   7544   7591   7638   7685   7732   7779   7827   7874   7921   7968   8015   8062
KMCSRH  002720       3984#  4106*  4107*  4121*  4123*  4124*  4580*  4581*
KMCTL   002722       3985#  4583*  4584*  4957
KMNUM   002612       3919#
KMPO4   002724       3986#  4265*  4279   4299*  4586*  4587*  4991
KMPO6   002726       3987#  4122*  4589*  4590*  5031   6344   6345
KMRLVL  002710       3980#  4229*  4594*  4595*
```

```
KMRVEC  002706            3979#   4227*   4592*
KMTLVL  002714            3982#   4230*   4600*   4601*
KMTVEC  002712            3981#   4228*   4597*   4598*
LNUNIT  035126            8139#
LOCK    002442            3888#
LOE   = 040000 G          3852#
LOGDEV  002552            3902#   4565*   4571*   4572    4574    4669
LOKFLG  002676            3949#
LOOPBK  035323            8142#
LOT   = 000010 G          3852#
LTN.ED= 000114            8184#
L$ACP   002110 G          3747#
L$APT   002036 G          3747#
L$AU    011364 G          3747    4718#
L$AUT   002070 G          3747#
L$AUTO  011256 G          3747    4655#
L$CCP   002106 G          3747#
L$CLEA  011354 G          3747    4683#
L$CO    002032 G          3747#
L$DEPO  002011 G          3747#
L$DESC  002414 G          3747    3882#
L$DESP  002076 G          3747#
L$DEVP  002060 G          3747#
L$DISP  002132 G          3747    3770#
L$DLY   002116 G          3747#
L$DTP   002040 G          3747#
L$DTYP  002034 G          3747#
L$DU    011360 G          3747    4700#
L$DUT   002072 G          3747#
L$DVTY  003130 G          3747    4015#
L$EF    002052 G          3747#
L$ENVI  002044 G          3747#
L$ETP   002102 G          3747#
L$EXP1  002046 G          3747#
L$EXP4  002064 G          3747#
L$EXP5  002066 G          3747#
L$HARD  034640 G          3747    8121#
L$HIME  002120 G          3747#
L$HPCP  002016 G          3747#
L$HPTP  002022 G          3747#
L$HW    002364 G          3747    3793#
L$ICP   002104 G          3747#
L$INIT  010570 G          3747    4537#
L$LADP  002026 G          3747#
L$LAST  040004 G          3747    8183#
L$LOAD  002100 G          3747#
L$LUN   002074 G          3747#
L$MREV  002050 G          3747#
L$NAME  002000 G          3747#
L$PRIO  002042 G          3747#
L$PROT  002122 G          3747    3756#
L$PRT   002112 G          3747#
L$REPP  002062 G          3747#
L$REV   002010 G          3747#
L$RPT   010562 G          4506#
L$SOFT  035404 G          8164#
```

```
L$SPC   002056 G          3747#
L$SPCP  002020 G          3747#
L$SPTP  002024 G          3747#
L$STA   002030 G          3747#
L$SW    002414 G          3822#
L$TEST  002114 G          3747#
L$TIML  002014 G          3747#
L$UNIT  002012 G          3747#   4572
L10001  002412            3793    3808#
L10002  002414            3822    3825#
L10003  006130            4455#
L10004  006206            4456#
L10005  006264            4457#
L10006  006346            4458#
L10007  006430            4459#
L10010  006512            4460#
L10011  006574            4461#
L10012  006656            4462#
L10013  006734            4463#
L10014  007012            4464#
L10015  007040            4465#
L10016  007116            4467#
L10017  007174            4469#
L10020  007222            4470#
L10021  007250            4471#
L10022  007326            4472#
L10023  007354            4473#
L10024  007402            4474#
L10025  007460            4475#
L10026  007506            4476#
L10027  007564            4477#
L10030  007642            4478#
L10031  007724            4479#
L10032  010002            4480#
L10033  010064            4481#
L10034  010142            4482#
L10035  010170            4483#
L10036  010216            4484#
L10037  010274            4485#
L10040  010352            4486#
L10041  010430            4487#
L10042  010506            4488#
L10043  010560            4493#
L10044  010566            4514    4523#
L10045  011254            4652#
L10046  011352            4673#
L10047  011356            4686#
L10050  011362            4703#
L10051  011364            4719#
L10052  011512            4744    4755    4756#
L10053  011556            4770#
L10054  011744            4822#
L10055  012110            4852#
L10056  012244            4888#
L10057  012374            4916#
L10060  012524            4944#
```

```
L10061  012666          4974#
L10062  013052          5020#
L10063  013236          5061#
L10064  013424          5100#
L10065  013574          5131#
L10066  013702          5158    5160#
L10067  014132          5210#
L10070  014362          5259#
L10071  014612          5308#
L10072  015042          5357#
L10073  015336          5416#
L10074  015632          5477#
L10075  016062          5526#
L10076  016312          5575#
L10077  016542          5624#
L10100  016772          5673#
L10101  017222          5722#
L10102  017452          5771#
L10103  017702          5820#
L10104  020132          5869#
L10105  020430          5926#
L10106  020660          5978#
L10107  021224          6034    6045#
L10110  021222          6044#
L10111  021536          6105#
L10112  021700          6131#
L10113  022042          6156#
L10114  022216          6193#
L10115  022422          6231#
L10116  022566          6262    6263    6266#
L10117  022736          6296    6297    6300#
L10120  023102          6329    6330    6333#
L10121  023270          6383#
L10122  023504          6427#
L10123  023722          6472#
L10124  024054          6504#
L10125  024272          6539    6553    6557#
L10126  024422          6592#
L10127  024636          6637    6641#
L10130  025042          6683    6687#
L10131  025246          6729    6733#
L10132  025452          6775    6779#
L10133  025656          6821    6825#
L10134  026062          6867    6871#
L10135  026266          6913    6917#
L10136  026472          6959    6963#
L10137  026700          7005    7010#
L10140  027104          7053    7057#
L10141  027310          7101    7106#
L10142  027514          7149    7153#
L10143  027720          7196    7200#
L10144  030124          7243    7247#
L10145  030330          7290    7294#
L10146  030534          7338    7342#
L10147  030740          7385    7389#
L10150  031144          7434    7438#
```

```
L10151  031350          7481    7485#
L10152  031554          7528    7532#
L10153  031760          7575    7579#
L10154  032164          7622    7626#
L10155  032370          7669    7673#
L10156  032574          7716    7720#
L10157  033000          7763    7767#
L10160  033204          7810    7814#
L10161  033410          7858    7862#
L10162  033614          7905    7909#
L10163  034020          7952    7956#
L10164  034224          7999    8003#
L10165  034430          8046    8050#
L10166  034634          8093    8097#
L10167  034704          8121    8132#
L10170  035404          8164    8167#
MASKX   002572          3910#
MDATA   002506          3895#
MEMDAT  002654          3940#   6611    6657    6703    6749    6795    6841    6887    6933    6979    7027    7075    7123
                        7170    7217    7264    7312    7359    7408    7455    7502    7549    7596    7643    7690    7737
                        7784    7832    7879    7926    7973    8020    8067
MEMEND  037776          3915    8182#
MEMLD   003640          4252#   6571    6610    6656    6702    6748    6794    6840    6886    6932    6978    7026    7074
                        7122    7169    7216    7263    7311    7358    7407    7454    7501    7548    7595    7642    7689
                        7736    7783    7831    7878    7925    7972    8019    8066
MEMLD2  003650          4259#
MEMLIM  002604          3915#   6547
MEMSET  003424          4188#
MEMSZ   002606          3916#   4639*   4647*
MRO     002624          3924#   4282    4479    4481    4751    4768    4801    4815    4832*   4841    4847    4876    4884
                        4905    4913    4933    4941    4953*   4963    4969    4989*   4999    5013    5029*   5039    5054
                        5079    5093    5119    5128    5156    5170*   5220*   5269*   5318*   5367*   5426*   5487*   5536*
                        5585*   5634*   5683*   5732*   5781*   5830*   6126    6128    6151    6153    6190    6223    6227
                        6261    6295    6328    6381    6422    6459    6469    6502    6538    6589    6630    6676    6722
                        6768    6814    6860    6906    6952    6998    7046    7094    7142    7189    7236    7283    7331
                        7378    7427    7474    7521    7568    7615    7662    7709    7756    7803    7851    7898    7945
                        7992    8039    8086
NEWST   010704          4552    4555    4563#   4573
NEXT    002440          3887#
NPRSET  003574          4234#   6244    6277    6311    6346    6403    6439    6483    6524
ONE     002602          3914#
O$APTS= 000000          3706#   3747
O$AU  = 000001          3706#   3737#   3747
O$BGNR= 000000          3706#   3747
O$BGNS= 000000          3706#   3747
O$DU  = 000001          3706#   3737#   3747
O$ERRT= 000000          3706#   3747
O$GNSW= 000000          3706#   3747
O$POIN= 000001          3706#   3737#   3747
O$SETU= 000000          3706#   3747    8183
PNT   = 001000 G        3852#
POP2SP- 022626          3861#
PRI   = 002000 G        3852#
PRIRTY  035064          8126    8138#
PRI00 = 000000 G        3852#   6124    6149
PRI01 = 000040 G        3852#
```

```
PRI02 = 000100 G          3852#
PRI03 = 000140 G          3852#
PRI04 = 000200 G          3852#
PRI05 = 000240 G          3852#
PRI06 = 000300 G          3852#
PRI07 = 000340 G          3852#  6120    6145
PSTACK  002554            3903#  4542*
QV.FLG  002677            3950#
RAMDAT  003370            4172#
REGADR  002730            3991#
RETADR  002562            3906#
RUN     002622            3923#
SAVACT  002614            3920#
SAVE4   002650            3934#  4545*   4548    4671    4753    6426
SAVE6   002652            3935#  4546*   4549    4672    4754    6425
SAVNUM  002616            3921#
SAVPC   002576            3912#
SAVSP   002574            3911#
SETBR0  003320            4137#
SETBR1  003330            4143#
SETBR4  003340            4150#
SETBR7  003350            4157#
SETC    004076            4320#  7363    7412    7459    7506    7553    7600    7647    7694    7741    7788    7836    7883
                          7930   7977    8024    8071
SETVEC  003552            4224#  6116    6141    6176    6214
SETZ    003360            4165#
SFPTBL  002414 G          3822#
SPDAT   002664            3941#  6613    6659    6705    6751    6797    6843    6889    6935    6981    7029    7077    7125
                          7172   7219    7266    7314    7361    7410    7457    7504    7551    7598    7645    7692    7739
                          7786   7834    7881    7928    7975    8022    8069
SPLD    004012            4292#  6573    6612    6658    6704    6750    6796    6842    6888    6934    6980    7028    7076
                          7124   7171    7218    7265    7313    7360    7409    7456    7503    7550    7597    7644    7691
                          7738   7785    7833    7880    7927    7974    8021    8068
SSTACK  003130            3995#  4540
STAT    002566            3908#
STAT1   002700            3972#  4603*   4605*   4609*   4620*   4624*   4627*   6169    6206
STAT2   002702            3973#  4630*   4632*
STAT3   002704            3974#
STOP    023302            6398#
STRTSW  002564            3907#
SUBRPC  002556            3904#
SVCGBL- 000000            3706#  3712    3720#   3747    3756    3770    3793    3822    3882    4015    4455    4456    4457
                          4458   4459    4460    4461    4462    4463    4464    4465    4467    4469    4470    4471    4472
                          4473   4474    4475    4476    4477    4478    4479    4480    4481    4482    4483    4484    4485
                          4486   4487    4488    4490    4506    4537    4655    4683    4700    4718    8121    8164    8183#
SVCINS- 000000            3706#  3717#   3747    3770    3793    3822    3882    4015    4282    4455    4456    4457    4458
                          4459   4460    4461    4462    4463    4464    4465    4467    4469    4470    4471    4472    4473
                          4474   4475    4476    4477    4478    4479    4480    4481    4482    4483    4484    4485    4486
                          4487   4488    4491    4492    4493    4514    4523    4551    4552    4554    4555    4557    4558
                          4561   4562    4574    4575    4652    4669    4673    4684    4686    4702    4703    4719    4744
                          4751   4755    4756    4768    4770    4795    4801    4802    4807    4810    4815    4816    4821
                          4822   4834    4841    4842    4847    4848    4851    4852    4869    4876    4877    4878    4879
                          4884   4885    4886    4888    4898    4905    4906    4907    4908    4913    4914    4916    4926
                          4933   4934    4935    4936    4941    4942    4944    4955    4963    4964    4969    4970    4973
                          4974   4993    4999    5000    5004    5006    5013    5014    5019    5020    5033    5039    5040
                          5044   5046    5054    5055    5060    5061    5073    5079    5080    5087    5088    5093    5094
```

CROSS REFERENCE TABLE -- USER SYMBOLS

5099 5100 5112 5119 5120 5121 5123 5128 5129 5130 5131 5156 5158
5160 5174 5184 5185 5139 5192 5203 5204 5209 5210 5223 5233 5234
5238 5241 5252 5253 5258 5259 5272 5282 5283 5287 5290 5301 5302
5307 5308 5321 5331 5332 5336 5339 5350 5351 5356 5357 5370 5384
5385 5390 5393 5408 5409 5415 5416 5429 5443 5445 5450 5453 5469
5470 5476 5477 5490 5500 5501 5505 5508 5519 5520 5525 5526 5539
5549 5550 5554 5557 5568 5569 5574 5575 5588 5598 5599 5603 5606
5617 5618 5623 5624 5637 5647 5648 5652 5655 5666 5667 5672 5673
5686 5696 5697 5701 5704 5715 5716 5721 5722 5735 5745 5746 5750
5753 5764 5765 5770 5771 5784 5794 5795 5799 5802 5813 5814 5819
5820 5833 5843 5844 5848 5851 5862 5863 5868 5869 5882 5900 5901
5906 5909 5918 5919 5925 5926 5939 5949 5950 5954 5957 5969 5970
5977 5978 5991 5992 6008 6009 6013 6015 6033 6034 6039 6044 6045
6058 6077 6078 6083 6086 6098 6099 6104 6105 6114 6120 6124 6126
6128 6131 6145 6149 6151 6153 6156 6168 6183 6189 6190 6193 6205
6221 6223 6227 6231 6239 6261 6262 6263 6266 6295 6296 6297 6300
6328 6329 6330 6333 6381 6383 6422 6427 6459 6469 6472 6502 6504
6538 6539 6553 6557 6575 6589 6590 6591 6592 6614 6630 6631 6636
6637 6641 6660 6676 6677 6682 6683 6687 6706 6722 6723 6728 6729
6733 6752 6768 6769 6774 6775 6779 6798 6814 6815 6820 6821 6825
6844 6860 6861 6866 6867 6871 6890 6906 6907 6912 6913 6917 6936
6952 6953 6958 6959 6963 6982 6998 6999 7004 7005 7010 7030 7046
7047 7052 7053 7057 7078 7094 7095 7100 7101 7106 7126 7142 7143
7148 7149 7153 7173 7189 7190 7195 7196 7200 7220 7236 7237 7242
7243 7247 7267 7283 7284 7289 7290 7294 7315 7331 7332 7337 7338
7342 7362 7378 7379 7384 7385 7389 7411 7427 7428 7433 7434 7438
7458 7474 7475 7480 7481 7485 7505 7521 7522 7527 7528 7532 7552
7568 7569 7574 7575 7579 7599 7615 7616 7621 7622 7626 7646 7662
7663 7668 7669 7673 7693 7709 7710 7715 7716 7720 7740 7756 7757
7762 7763 7767 7787 7803 7804 7809 7810 7814 7835 7851 7852 7857
7858 7862 7882 7898 7899 7904 7905 7909 7929 7945 7946 7951 7952
7956 7976 7992 7993 7998 7999 8003 8023 8039 8040 8045 8046 8050
8070 8086 8087 8092 8093 8097 8121 8123 8124 8125 8126 8132 8164
8167 8183

SVCSUB= 000000 3706# 3719# 5991

SVCTAG= 000000 3706# 3721# 3808 3825 4455 4456 4457 4458 4459 4460 4461 4462 4463
4464 4465 4467 4469 4470 4471 4472 4473 4474 4475 4476 4477 4478
4479 4480 4481 4482 4483 4484 4485 4486 4487 4488 4493 4523 4652
4673 4686 4703 4719 4756 4770 4807 4821 4822 4851 4852 4878 4886
4888 4907 4914 4916 4935 4942 4944 4973 4974 5004 5019 5020 5044
5060 5061 5087 5099 5100 5121 5130 5131 5160 5189 5209 5210 5238
5258 5259 5287 5307 5308 5336 5356 5357 5390 5415 5416 5450 5476
5477 5505 5525 5526 5554 5574 5575 5603 5623 5624 5652 5672 5673
5701 5721 5722 5750 5770 5771 5799 5819 5820 5848 5868 5869 5906
5925 5926 5954 5977 5978 6013 6039 6044 6045 6083 6104 6105 6131
6156 6193 6231 6266 6300 6333 6383 6427 6472 6504 6557 6591 6592
6636 6641 6682 6687 6728 6733 6774 6779 6820 6825 6866 6871 6912
6917 6958 6963 7004 7010 7052 7057 7100 7106 7148 7153 7195 7200
7242 7247 7289 7294 7337 7342 7384 7389 7433 7438 7480 7485 7527
7532 7574 7579 7621 7626 7668 7673 7715 7720 7762 7767 7809 7814
7857 7862 7904 7909 7951 7956 7998 8003 8045 8050 8092 8097 8132
8167

SVCTST- 000000 3706# 3718# 4737 4762 4790 4830 4867 4896 4924 4951 4988 5028 5069
5108 5144 5168 5218 5267 5316 5365 5424 5485 5534 5583 5632 5681
5730 5779 5828 5877 5934 5986 6053 6112 6138 6164 6201 6238 6273
6307 6341 6395 6436 6480 6512 6568 6605 6651 6697 6743 6789 6835

```
                        6881  6927  6973  7021  7068  7117  7164  7211  7258  7306  7353  7402  7449
                        7496  7543  7590  7637  7684  7731  7778  7826  7873  7920  7967  8013  8061
SWPAC1  035217          8140#
SWPAC2  035260          8141#
S$LSYM= 010000          3706#  3808#  3825#  4455#  4456#  4457#  4458#  4459#  4460#  4461#  4462#  4463#  4464#
                        4465#  4467#  4469#  4470#  4471#  4472#  4473#  4474#  4475#  4476#  4477#  4478#  4479#
                        4480#  4481#  4482#  4483#  4484#  4485#  4486#  4487#  4488#  4493#  4523#  4652#  4673#
                        4686#  4703#  4719#  4756#  4770#  4795#  4810#  4822#  4834#  4852#  4869#  4879#  4888#
                        4898#  4908#  4916#  4926#  4936#  4944#  4955#  4974#  4993#  5006#  5020#  5033#  5046#
                        5061#  5073#  5088#  5100#  5112#  5123#  5131#  5160#  5174#  5192#  5210#  5223#  5241#
                        5259#  5272#  5290#  5308#  5321#  5339#  5357#  5370#  5393#  5416#  5429#  5453#  5477#
                        5490#  5508#  5526#  5539#  5557#  5575#  5588#  5606#  5624#  5637#  5655#  5673#  5686#
                        5704#  5722#  5735#  5753#  5771#  5784#  5802#  5820#  5833#  5851#  5869#  5882#  5909#
                        5926#  5939#  5957#  5978#  5992#  6015#  6044#  6045#  6058#  6086#  6105#  6131#  6156#
                        6193#  6231#  6266#  6300#  6333#  6383#  6427#  6472#  6504#  6557#  6575#  6592#  6614#
                        6641#  6660#  6687#  6706#  6733#  6752#  6779#  6798#  6825#  6844#  6871#  6890#  6917#
                        6936#  6963#  6982#  7010#  7030#  7057#  7078#  7106#  7126#  7153#  7173#  7200#  7220#
                        7247#  7267#  7294#  7315#  7342#  7362#  7389#  7411#  7438#  7458#  7485#  7505#  7532#
                        7552#  7579#  7599#  7626#  7646#  7673#  7693#  7720#  7740#  7767#  7787#  7814#  7835#
                        7862#  7882#  7909#  7929#  7956#  7976#  8003#  8023#  8050#  8070#  8097#  8132#  8167#
TDATA   024424          6572   6574   6593#
TEMP    002444          3893#
TFM1    004123          4344#  4458   4459   4460   4461   4462
TFM2    004146          4345#  4455   4456   4457   4463   4464   4467   4472   4477   4478   4480   4482   4485
                        4486   4487   4488
TFM27   004200          4347#  4479   4481
TFM37   004225          4348#  4492
TFM5    004163          4346#  4469   4475
TOUTP   023470          6421   6425#
TOUTT   023464          6398   6424#
TYPE    002630          3926#  4640*  4645*  4650*
T$ARGC- 000002          3747#  4455#  4456#  4457#  4458#  4459#  4460#  4461#  4462#  4463#  4464#  4465#  4467#
                        4469#  4470#  4471#  4472#  4473#  4474#  4475#  4476#  4477#  4478#  4479#  4480#  4481#
                        4482#  4483#  4484#  4485#  4486#  4487#  4488#  4491#  4492#
T$CODE- 003032          8123#  8124#  8125#  8126#
T$ERRN  000027          3706#  4282#  4751#  4768#  4801#  4815#  4841#  4847#  4876#  4884#  4905#  4913#  4933#
                        4941#  4963#  4969#  4999#  5013#  5039#  5054#  5079#  5093#  5119#  5128#  5156#  5184#
                        5203#  5233#  5252#  5282#  5301#  5331#  5350#  5384#  5408#  5443#  5469#  5500#  5519#
                        5549#  5568#  5598#  5617#  5647#  5666#  5696#  5715#  5745#  5764#  5794#  5813#  5843#
                        5862#  5900#  5918#  5949#  5969#  6008#  6033#  6077#  6098#  6126#  6128#  6151#  6153#
                        6190#  6223#  6227#  6261#  6295#  6328#  6381#  6422#  6459#  6469#  6502#  6538#  6589#
                        6630#  6676#  6722#  6768#  6814#  6860#  6906#  6952#  6998#  7046#  7094#  7142#  7189#
                        7236#  7283#  7331#  7378#  7427#  7474#  7521#  7568#  7615#  7662#  7709#  7756#  7803#
                        7851#  7898#  7945#  7992#  8039#  8086#
T$EXCP- 000000          8123#  8124#  8125#  8126#
T$FLAG- 000040          4514#  4744#  4755#  4802#  4816#  4842#  4848#  4877#  4885#  4906#  4934#  4964#  4970#
                        5000#  5014#  5040#  5055#  5080#  5094#  5120#  5129#  5158#  5185#  5204#  5234#  5253#
                        5283#  5302#  5332#  5351#  5385#  5409#  5445#  5470#  5501#  5520#  5550#  5569#  5599#
                        5618#  5648#  5667#  5697#  5716#  5746#  5765#  5795#  5814#  5844#  5863#  5901#  5919#
                        5950#  5970#  6009#  6034#  6078#  6099#  6262#  6263#  6296#  6297#  6329#  6330#  6539#
                        6553#  6590#  6631#  6637#  6677#  6683#  6723#  6729#  6769#  6775#  6815#  6821#  6861#
                        6867#  6907#  6913#  6953#  6959#  6999#  7005#  7047#  7053#  7095#  7101#  7143#  7149#
                        7190#  7196#  7237#  7243#  7284#  7290#  7332#  7338#  7379#  7385#  7428#  7434#  7475#
                        7481#  7522#  7528#  7569#  7575#  7616#  7622#  7663#  7669#  7710#  7716#  7757#  7763#
                        7804#  7810#  7852#  7858#  7899#  7905#  7946#  7952#  7993#  7999#  8040#  8046#  8087#
                        8093#
```

T$GMAN= 000000 3706#
T$HILI= 000007 8123# 8124# 8125# 8126#
T$LAST= 000001 3706# 8183#
T$LOLI= 000004 8123# 8124# 8125# 8126#
T$LSYM= 010000 3706# 3808 3825 4455 4456 4457 4458 4459 4460 4461 4462 4463 4464
4465 4467 4469 4470 4471 4472 4473 4474 4475 4476 4477 4478 4479
4480 4481 4482 4483 4484 4485 4486 4487 4488 4493 4523 4652 4673
4686 4703 4719 4756 4770 4822 4852 4888 4916 4944 4974 5020 5061
5100 5131 5160 5210 5259 5308 5357 5416 5477 5526 5575 5624 5673
5722 5771 5820 5869 5926 5978 6044 6045 6105 6131 6156 6193 6231
6266 6300 6333 6383 6427 6472 6504 6557 6592 6641 6687 6733 6779
6825 6871 6917 6963 7010 7057 7106 7153 7200 7247 7294 7342 7389
7438 7485 7532 7579 7626 7673 7720 7767 7814 7862 7909 7956 8003
8050 8097 8132 8167
T$LTNC= 000114 8183#
T$NEST= 177777 3706# 3712# 3756# 3760# 3793# 3808# 3822# 3825# 4455# 4456# 4457# 4458# 4459#
4460# 4461# 4462# 4463# 4464# 4465# 4467# 4469# 4470# 4471# 4472# 4473# 4474#
4475# 4476# 4477# 4478# 4479# 4480# 4481# 4482# 4483# 4484# 4485# 4486# 4487#
4488# 4490# 4493# 4506# 4523# 4537# 4652# 4655# 4673# 4683# 4686# 4700# 4703#
4718# 4719# 4737# 4756# 4762# 4770# 4790# 4795# 4807# 4810# 4821# 4822# 4830#
4834# 4851# 4852# 4867# 4869# 4878# 4879# 4886# 4888# 4896# 4898# 4907# 4908#
4914# 4916# 4924# 4926# 4935# 4936# 4942# 4944# 4951# 4955# 4973# 4974# 4988#
4993# 5004# 5006# 5019# 5020# 5028# 5033# 5044# 5046# 5060# 5061# 5069# 5073#
5087# 5088# 5099# 5100# 5108# 5112# 5121# 5123# 5130# 5131# 5144# 5160# 5168#
5174# 5189# 5192# 5209# 5210# 5218# 5223# 5238# 5241# 5258# 5259# 5267# 5272#
5287# 5290# 5307# 5308# 5316# 5321# 5336# 5339# 5356# 5357# 5365# 5370# 5390#
5393# 5415# 5416# 5424# 5429# 5450# 5453# 5476# 5477# 5485# 5490# 5505# 5508#
5525# 5526# 5534# 5539# 5554# 5557# 5574# 5575# 5583# 5588# 5603# 5606# 5623#
5624# 5632# 5637# 5652# 5655# 5672# 5673# 5681# 5686# 5701# 5704# 5721# 5722#
5730# 5735# 5750# 5753# 5770# 5771# 5779# 5784# 5799# 5802# 5819# 5820# 5828#
5833# 5848# 5851# 5868# 5869# 5877# 5882# 5906# 5909# 5925# 5926# 5934# 5939#
5954# 5957# 5977# 5978# 5986# 5991# 5992# 6013# 6015# 6039# 6044# 6045# 6053#
6058# 6083# 6086# 6104# 6105# 6112# 6131# 6138# 6156# 6164# 6193# 6201# 6231#
6238# 6266# 6273# 6300# 6307# 6333# 6341# 6383# 6395# 6427# 6436# 6472# 6480#
6504# 6512# 6557# 6568# 6575# 6591# 6592# 6605# 6614# 6636# 6641# 6651# 6660#
6682# 6687# 6697# 6706# 6728# 6733# 6743# 6752# 6774# 6779# 6789# 6798# 6820#
6825# 6835# 6844# 6866# 6871# 6881# 6890# 6912# 6917# 6927# 6936# 6958# 6963#
6973# 6982# 7004# 7010# 7021# 7030# 7052# 7057# 7068# 7078# 7100# 7106# 7117#
7126# 7148# 7153# 7164# 7173# 7195# 7200# 7211# 7220# 7242# 7247# 7258# 7267#
7289# 7294# 7306# 7315# 7337# 7342# 7353# 7362# 7384# 7389# 7402# 7411# 7433#
7438# 7449# 7458# 7480# 7485# 7496# 7505# 7527# 7532# 7543# 7552# 7574# 7579#
7590# 7599# 7621# 7626# 7637# 7646# 7668# 7673# 7684# 7693# 7715# 7720# 7731#
7740# 7762# 7767# 7778# 7787# 7809# 7814# 7826# 7835# 7857# 7862# 7873# 7882#
7904# 7909# 7920# 7929# 7951# 7956# 7967# 7976# 7998# 8003# 8013# 8023# 8045#
8050# 8061# 8070# 8092# 8097# 8121# 8132# 8164# 8167# 8175#
T$NS0 - 000000 3712# 8175
T$NS1 - 000005 3756# 3760 3793# 3808 3822# 3825 4455# 4456# 4457# 4458# 4459# 4460# 4461#
4462# 4463# 4464# 4465# 4467# 4469# 4470# 4471# 4472# 4473# 4474# 4475# 4476#
4477# 4478# 4479# 4480# 4481# 4482# 4483# 4484# 4485# 4486# 4487# 4488# 4490#
4493 4506# 4523 4537# 4652 4655# 4673 4683# 4686 4700# 4703 4718# 4719
4737# 4756 4762# 4770 4790# 4822 4830# 4852 4867# 4888 4896# 4916 4924#
4944 4951# 4974 4988# 5020 5028# 5061 5069# 5100 5108# 5131 5144# 5160
5168# 5210 5218# 5259 5267# 5308 5316# 5357 5365# 5416 5424# 5477 5485#
5526 5534# 5575 5583# 5624 5632# 5673 5681# 5722 5730# 5771 5779# 5820
5828# 5869 5877# 5926 5934# 5978 5986# 6045 6053# 6105 6112# 6131 6138#
6156 6164# 6193 6201# 6231 6238# 6266 6273# 6300 6307# 6333 6341# 6383

```
                    6395#  6427   6436#  6472   6480#  6504   6512#  6557   6568#  6592   6605#  6641   6651#
                    6687   6697#  6733   6743#  6779   6789#  6825   6835#  6871   6881#  6917   6927#  6963
                    6973#  7010   7021#  7057   7068#  7106   7117#  7153   7164#  7200   7211#  7247   7258#
                    7294   7306#  7342   7353#  7389   7402#  7438   7449#  7485   7496#  7532   7543#  7579
                    7590#  7626   7637#  7673   7684#  7720   7731#  7767   7778#  7814   7826#  7862   7873#
                    7909   7920#  7956   7967#  8003   8013#  8050   8061#  8097   8121#  8132   8164#  8167
T$NS2 = 000003       4795#  4807   4810#  4821   4834#  4851   4869#  4878   4879#  4886   4898#  4907   4908#
                    4914   4926#  4935   4936#  4942   4955#  4973   4993#  5004   5006#  5019   5033#  5044
                    5046#  5060   5073#  5087   5088#  5099   5112#  5121   5123#  5130   5174#  5189   5192#
                    5209   5223#  5238   5241#  5258   5272#  5287   5290#  5307   5321#  5336   5339#  5356
                    5370#  5390   5393#  5415   5429#  5450   5453#  5476   5490#  5505   5508#  5525   5539#
                    5554   5557#  5574   5588#  5603   5606#  5623   5637#  5652   5655#  5672   5686#  5701
                    5704#  5721   5735#  5750   5753#  5770   5784#  5799   5802#  5819   5833#  5848   5851#
                    5868   5882#  5906   5909#  5925   5939#  5954   5957#  5977   5991#  6044   6058#  6083
                    6086#  6104   6575#  6591   6614#  6636   6660#  6682   6706#  6728   6752#  6774   6798#
                    6820   6844#  6866   6890#  6912   6936#  6958   6982#  7004   7030#  7052   7078#  7100
                    7126#  7148   7173#  7195   7220#  7242   7267#  7289   7315#  7337   7362#  7384   7411#
                    7433   7458#  7480   7505#  7527   7552#  7574   7599#  7621   7646#  7668   7693#  7715
                    7740#  7762   7787#  7809   7835#  7857   7882#  7904   7929#  7951   7976#  7998   8023#
                    8045   8070#  8092
T$NS3 = 000003       5992#  6013   6015#  6039
T$PTNU= 000000       3706#
T$SAVL= 177777       3706#
T$SEGL= 177777       3706#  4795#  4802   4807#  4810#  4816   4821#  4834#  4842   4848   4851#  4869#  4877
                    4878#  4879#  4885   4886#  4898#  4906   4907#  4908#  4914#  4926#  4934   4935#  4936#
                    4942#  4955#  4964   4970   4973#  4993#  5000   5004#  5006#  5014   5019#  5033#  5040
                    5044#  5046#  5055   5060#  5073#  5080   5087#  5088#  5094   5099#  5112#  5120   5121#
                    5123#  5129   5130#  5174#  5185   5189#  5192#  5204   5209#  5223#  5234   5238#  5241#
                    5253   5258#  5272#  5283   5287#  5290#  5302   5307#  5321#  5332   5336#  5339#  5351
                    5356#  5370#  5385   5390#  5393#  5409   5415#  5429#  5445   5450#  5453#  5470   5476#
                    5490#  5501   5505#  5508#  5520   5525#  5539#  5550   5554#  5557#  5569   5574#  5588#
                    5599   5603#  5606#  5618   5623#  5637#  5648   5652#  5655#  5667   5672#  5686#  5697
                    5701#  5704#  5716   5721#  5735#  5746   5750#  5753#  5765   5770#  5784#  5795   5799#
                    5802#  5814   5819#  5833#  5844   5848#  5851#  5863   5868#  5882#  5901   5906#  5909#
                    5919   5925#  5939#  5950   5954#  5957#  5970   5977#  5992#  6009   6013#  6015#  6039#
                    6058#  6078   6083#  6086#  6099   6104#  6575#  6590   6591#  6614#  6631   6636#  6660#
                    6677   6682#  6706#  6723   6728#  6752#  6769   6774#  6798#  6815   6820#  6844#  6861
                    6866#  6890#  6907   6912#  6936#  6953   6958#  6982#  6999   7004#  7030#  7047   7052#
                    7078#  7095   7100#  7126#  7143   7148#  7173#  7190   7195#  7220#  7237   7242#  7267#
                    7284   7289#  7315#  7332   7337#  7362#  7379   7384#  7411#  7428   7433#  7458#  7475
                    7480#  7505#  7522   7527#  7552#  7569   7574#  7599#  7616   7621#  7646#  7663   7668#
                    7693#  7710   7715#  7740#  7757   7762#  7787#  7804   7809#  7835#  7852   7857#  7882#
                    7899   7904#  7929#  7946   7951#  7976#  7993   7998#  8023#  8040   8045#  8070#  8087
                    8092#
T$SEKO= 010000       4795#  4802   4807   4810#  4816   4821   4834#  4842   4848   4851   4869#  4877   4878
                    4879#  4885   4886   4898#  4906   4907   4908#  4914   4926#  4934   4935   4936#  4942
                    4955#  4964   4970   4973   4993#  5000   5004   5006#  5014   5019   5033#  5040   5044
                    5046#  5055   5060   5073#  5080   5087   5088#  5094   5099   5112#  5120   5121   5123#
                    5129   5130   5174#  5185   5189   5192#  5204   5209   5223#  5234   5238   5241#  5253
                    5258   5272#  5283   5287   5290#  5302   5307   5321#  5332   5336   5339#  5351   5356
                    5370#  5385   5390   5393#  5409   5415   5429#  5445   5450   5453#  5470   5476   5490#
                    5501   5505   5508#  5520   5525   5539#  5550   5554   5557#  5569   5574   5588#  5599
                    5603   5606#  5618   5623   5637#  5648   5652   5655#  5667   5672   5686#  5697   5701
                    5704#  5716   5721   5735#  5746   5750   5753#  5765   5770   5784#  5795   5799   5802#
                    5814   5819   5833#  5844   5848   5851#  5863   5868   5882#  5901   5906   5909#  5919
                    5925   5939#  5950   5954   5957#  5970   5977   5992#  6009   6013   6015#  6039   6058#
```

```
                    6078  6083  6086#  6099  6104  6575#  6590  6591  6614#  6631  6636  6660#  6677
                    6682  6706#  6723  6728  6752#  6769  6774  6798#  6815  6820  6844#  6861  6866
                    6890#  6907  6912  6936#  6953  6958  6982#  6999  7004  7030#  7047  7052  7078#
                    7095  7100  7126#  7143  7148  7173#  7190  7195  7220#  7237  7242  7267#  7284
                    7289  7315#  7332  7337  7362#  7379  7384  7411#  7428  7433  7458#  7475  7480
                    7505#  7522  7527  7552#  7569  7574  7599#  7616  7621  7646#  7663  7668  7693#
                    7710  7715  7740#  7757  7762  7787#  7804  7809  7835#  7852  7857  7882#  7899
                    7904  7929#  7946  7951  7976#  7993  7998  8023#  8040  8045  8070#  8087  8092
T$SUBN= 000000      3706#  4737#  4762#  4790#  4830#  4867#  4896#  4924#  495'#  4988#  5028#  5069#  5108#
                    5144#  5168#  5218#  5267#  5316#  5365#  5424#  5485#  5534#  5583#  5632#  5681#  5730#
                    5779#  5828#  5877#  5934#  5986#  5991#  6053#  6112#  6138#  6164#  6201#  6238#  6273#
                    6307#  6341#  6395#  6436#  6480#  6512#  6568#  6605#  6651#  6697#  6743#  6789#  6835#
                    6881#  6927#  6973#  7021#  7068#  7117#  7164#  7211#  7258#  7306#  7353#  7402#  7449#
                    7496#  7543#  7590#  7637#  7684#  7731#  7778#  7826#  7873#  7920#  7967#  8013#  8061#
T$TAGL= 177777      3706#
T$TAGN= 010171      3706#  3756#  3793#  3822#  4455#  4456#  4457#  4458#  4459#  4460#  4461#  4462#  4463#
                    4464#  4465#  4467#  4469#  4470#  4471#  4472#  4473#  4474#  4475#  4476#  4477#  4478#
                    4479#  4480#  4481#  4482#  4483#  4484#  4485#  4486#  4487#  4488#  4490#  4506#  4537#
                    4655#  4683#  4700#  4718#  4737#  4762#  4790#  4830#  4867#  4896#  4924#  4951#  4988#
                    5028#  5069#  5108#  5144#  5168#  5218#  5267#  5316#  5365#  5424#  5485#  5534#  5583#
                    5632#  5681#  5730#  5779#  5828#  5877#  5934#  5986#  5991#  6053#  6112#  6138#  6164#
                    6201#  6238#  6273#  6307#  6341#  6395#  6436#  6480#  6512#  6568#  6605#  6651#  6697#
                    6743#  6789#  6835#  6881#  6927#  6973#  7021#  7068#  7117#  7164#  7211#  7258#  7306#
                    7353#  7402#  7449#  7496#  7543#  7590#  7637#  7684#  7731#  7778#  7826#  7873#  7920#
                    7967#  8013#  8061#  8121#  8164#
T$TEMP= 000000      3760#  3770#  3808#  3825#  4455#  4456#  4457#  4458#  4459#  4460#  4461#  4462#  4463#
                    4464#  4465#  4467#  4469#  4470#  4471#  4472#  4473#  4474#  4475#  4476#  4477#  4478#
                    4479#  4480#  4481#  4482#  4483#  4484#  4485#  4486#  4487#  4488#  4493#  4514#  4523#
                    4652#  4673#  4686#  4703#  4719#  4744#  4755#  4756#  4770#  4802#  4807#  4816#  4821#
                    4822#  4842#  4848#  4851#  4852#  4877#  4878#  4885#  4886#  4888#  4906#  4907#  4914#
                    4916#  4934#  4935#  4942#  4944#  4964#  4970#  4973#  4974#  5000#  5004#  5014#  5019#
                    5020#  5040#  5044#  5055#  5060#  5061#  5080#  5087#  5094#  5099#  5100#  5120#  5121#
                    5129#  5130#  5131#  5153#  5160#  5185#  5189#  5204#  5209#  5210#  5234#  5238#  5253#
                    5258#  5259#  5283#  5287#  5302#  5307#  5308#  5332#  5336#  5351#  5356#  5357#  5385#
                    5390#  5409#  5415#  5416#  5445#  5450#  5-  0#  5476#  5477#  5501#  5505#  5520#  5525#
                    5526#  5550#  5554#  5569#  5574#  5575#  5599#  5603#  5618#  5623#  5624#  5648#  5652#
                    5667#  5672#  5673#  5697#  5701#  5716#  5721#  5722#  5746#  5750#  5765#  5770#  5771#
                    5795#  5799#  5814#  5819#  5820#  5844#  5848#  5863#  5868#  5869#  5901#  5906#  5919#
                    5925#  5926#  5950#  5954#  5970#  5977#  5978#  6009#  6013#  6034#  6039#  6044#  6045#
                    6078#  6083#  6099#  6104#  6105#  6131#  6156#  6193#  6231#  6262#  6263#  6266#  6296#
                    6297#  6300#  6329#  6330#  6333#  6383#  6427#  6472#  6504#  6539#  6553#  6557#  6590#
                    6591#  6592#  6631#  6636#  6637#  6641#  6677#  6682#  6683#  6687#  6723#  6728#  6729#
                    6733#  6769#  6774#  6775#  6779#  6815#  6820#  6821#  6825#  6861#  6866#  6867#  6871#
                    6907#  6912#  6913#  6917#  6953#  6958#  6959#  6963#  6999#  7004#  7005#  7010#  7047#
                    7052#  7053#  7057#  7095#  7100#  7101#  7106#  7143#  7148#  7149#  7153#  7190#  7195#
                    7196#  7200#  7237#  7242#  7243#  7247#  7284#  7289#  7290#  7294#  7332#  7337#  7338#
                    7342#  7379#  7384#  7385#  7389#  7428#  7433#  7434#  7438#  7475#  7480#  7481#  7485#
                    7522#  7527#  7528#  7532#  7569#  7574#  7575#  7579#  7616#  7621#  7622#  7626#  7663#
                    7668#  7669#  7673#  7710#  7715#  7716#  7720#  7757#  7762#  7763#  7767#  7804#  7809#
                    7810#  7814#  7852#  7857#  7858#  7862#  7899#  7904#  7905#  7909#  7946#  7951#  7952#
                    7956#  7993#  7998#  7999#  8003#  8040#  8045#  8046#  8050#  8087#  8092#  8093#  8097#
                    8123#  8124#  8125#  8126#  8132#  8167#  8175#
T$TEST= 000114      3706#  4732  4735  4737#  4758  4760  4762#  4772  4788  4790#  4824  482°  4830#
                    4861  4865  4867#  4890  4894  4896#  4918  4922  4924#  4946  4949  4951#  4981
                    4985  4988#  5022  5026  5028#  5063  5067  5069#  5102  5106  5108#  5136  5142
                    5144#  5162  5166  5168#  5212  5216  5218#  5261  5265  5267#  5310  5314  5316#
```

	5359	5363	5365#	5418	5422	5424#	5479	5483	5485#	5528	5532	5534#	5577
	5581	5583#	5626	5630	5632#	5675	5679	5681#	5724	5728	5730#	5773	5777
	5779#	5822	5826	5828#	5871	5875	5877#	5928	5932	5934#	5980	5984	5986#
	5991	6047	6051	6053#	6107	6110	6112#	6133	6136	6138#	6158	6162	6164#
	6195	6199	6201#	6233	6236	6238#	6268	6271	6273#	6302	6305	6307#	6335
	6339	6341#	6385	6393	6395#	6429	6434	6436#	6474	6478	6480#	6506	6510
	6512#	6563	6566	6568#	6597	6603	6605#	6643	6649	6651#	6689	6695	6697#
	6735	6741	6743#	6781	6787	6789#	6827	6833	6835#	6873	6879	6881#	6919
	6925	6927#	6965	6971	6973#	7013	7019	7021#	7060	7066	7068#	7109	7115
	7117#	7156	7162	7164#	7203	7209	7211#	7250	7256	7258#	7297	7304	7306#
	7345	7351	7353#	7392	7398	7402#	7441	7447	7449#	7488	7494	7496#	7535
	7541	7543#	7582	7589	7590#	7629	7635	7637#	7676	7682	7684#	7723	7729
	7731#	7770	7776	7778#	7817	7823	7826#	7865	7871	7873#	7912	7918	7920#
	7959	7965	7967#	8005	8011	8013#	8053	8059	8061#	8183	8184		
T$TSTM= 177777	3706#	4282	4455	4456	4457	4458	4459	4460	4461	4462	4463	4464	4465
	4467	4469	4470	4471	4472	4473	4474	4475	4476	4477	4478	4479	4480
	4481	4482	4483	4484	4485	4486	4487	4488	4491	4492	4493	4523	4551
	4554	4557	4561	4574	4652	4669	4673	4684	4686	4702	4703	4719	4744
	4751	4755	4756	4768	4770	4795	4801	4802	4807	4810	4815	4816	4821
	4822	4834	4841	4842	4847	4848	4851	4852	4869	4876	4877	4878	4879
	4884	4885	4886	4888	4898	4905	4906	4907	4908	4913	4914	4916	4926
	4933	4934	4935	4936	4941	4942	4944	4955	4963	4964	4969	4970	4973
	4974	4993	4999	5000	5004	5006	5013	5014	5019	5020	5033	5039	5040
	5044	5046	5054	5055	5060	5061	5073	5079	5080	5087	5088	5093	5094
	5099	5100	5112	5119	5120	5121	5123	5128	5129	5130	5131	5156	5158
	5160	5174	5184	5185	5189	5192	5203	5204	5209	5210	5223	5233	5234
	5238	5241	5252	5253	5258	5259	5272	5282	5283	5287	5290	5301	5302
	5307	5308	5321	5331	5332	5336	5339	5350	5351	5356	5357	5370	5384
	5385	5390	5393	5408	5409	5415	5416	5429	5443	5445	5450	5453	5469
	5470	5476	5477	5490	5500	5501	5505	5508	5519	5520	5525	5526	5539
	5549	5550	5554	5557	5568	5569	5574	5575	5588	5598	5599	5603	5606
	5617	5618	5623	5624	5637	5647	5648	5652	5655	5666	5667	5672	5673
	5686	5696	5697	5701	5704	5715	5716	5721	5722	5735	5745	5746	5750
	5753	5764	5765	5770	5771	5784	5794	5795	5799	5802	5813	5814	5819
	5820	5833	5843	5844	5848	5851	5862	5863	5868	5869	5882	5900	5901
	5906	5909	5918	5919	5925	5926	5939	5949	5950	5954	5957	5969	5970
	5977	5978	5991	5992	6008	6009	6013	6015	6033	6034	6039	6044	6045
	6058	6077	6078	6083	6086	6098	6099	6104	6105	6114	6120	6124	6126
	6128	6131	6145	6149	6151	6153	6156	6168	6183	6189	6190	6193	6205
	6221	6223	6227	6231	6239	6261	6262	6263	6266	6295	6296	6297	6300
	6328	6329	6330	6333	6381	6383	6422	6427	6459	6469	6472	6502	6504
	6538	6539	6553	6557	6575	6589	6590	6591	6592	6614	6630	6631	6636
	6637	6641	6660	6676	6677	6682	6683	6687	6706	6722	6723	6728	6729
	6733	6752	6768	6769	6774	6775	6779	6798	6814	6815	6820	6821	6825
	6844	6860	6861	6866	6867	6871	6890	6906	6907	6912	6913	6917	6936
	6952	6953	6958	6959	6963	6982	6998	6999	7004	7005	7010	7030	7046
	7047	7052	7053	7057	7078	7094	7095	7100	7101	7106	7126	7142	7143
	7148	7149	7153	7173	7189	7190	7195	7196	7200	7220	7236	7237	7242
	7243	7247	7267	7283	7284	7289	7290	7294	7315	7331	7332	7337	7338
	7342	7362	7378	7379	7384	7385	7389	7411	7427	7428	7433	7434	7438
	7458	7474	7475	7480	7481	7485	7505	7521	7522	7527	7528	7532	7552
	7568	7569	7574	7575	7579	7599	7615	7616	7621	7622	7626	7646	7662
	7663	7668	7669	7673	7693	7709	7710	7715	7716	7720	7740	7756	7757
	7762	7763	7767	7787	7803	7804	7809	7810	7814	7835	7851	7852	7857
	7858	7862	7882	7898	7899	7904	7905	7909	7929	7945	7946	7951	7952
	7956	7976	7992	7993	7998	7999	8003	8023	8039	8040	8045	8046	8050

```
                    8070  8086  8087  8092  8093  8097
T$TSTS= 000001      3706# 4737# 4762# 4790# 4830# 4867# 4896# 4924# 4951# 4988# 5028# 5069# 5108#
                    5144# 5168# 5218# 5267# 5316# 5365# 5424# 5485# 5534# 5583# 5632# 5681# 5730#
                    5779# 5828# 5877# 5934# 5986# 6053# 6112# 6138# 6164# 6201# 6238# 6273# 6307#
                    6341# 6395# 6436# 6480# 6512# 6568# 6605# 6651# 6697# 6743# 6789# 6835# 6881#
                    6927# 6973# 7021# 7068# 7117# 7164# 7211# 7258# 7306# 7353# 7402# 7449# 7496#
                    7543# 7590# 7637# 7684# 7731# 7778# 7826# 7873# 7920# 7967# 8013# 8061#
T$$AU = 010051      4718# 4719
T$$AUT= 010046      4655# 4673
T$$CLE= 010047      4683# 4686
T$$DU = 010050      4700# 4703
T$$HAR= 010167      8121# 8132
T$$HW = 010001      3793# 3808
T$$INI= 010045      4537# 4652
T$$MSG= 010043      4455# 4456# 4457# 4458# 4459# 4460# 4461# 4462# 4463# 4464# 4465# 4467# 4469#
                    4470# 4471# 4472# 4473# 4474# 4475# 4476# 4477# 4478# 4479# 4480# 4481# 4482#
                    4483# 4484# 4485# 4486# 4487# 4488# 4490# 4493
T$$PRO= 010000      3756#
T$$RPT= 010044      4506# 4514  4523
T$$SEG= 010000      4795# 4802  4807# 4810# 4816  4821# 4834# 4842  4848  4851# 4869# 4877  4878#
                    4879# 4885  4886# 4898# 4906  4907# 4908# 4914# 4926# 4934  4935# 4936# 4942#
                    4955# 4964  4970  4973# 4993# 5000  5004# 5006# 5014  5019# 5033# 5040  5044#
                    5046# 5055  5060# 5073# 5080  5087# 5088# 5094  5099# 5112# 5120  5121# 5123#
                    5129  5130# 5174# 5185  5189# 5192# 5204  5209# 5223# 5234  5238# 5241# 5253
                    5258# 5272# 5283  5287# 5290# 5302  5307# 5321# 5332  5336# 5339# 5351  5356#
                    5370# 5385  5390# 5393# 5409  5415# 5429# 5445  5450# 5453# 5470  5476# 5490#
                    5501  5505# 5508# 5520  5525# 5539# 5550  5554# 5557# 5569  5574# 5588# 5599
                    5603# 5606# 5618  5623# 5637# 5648  5652# 5655# 5667  5672# 5686# 5697  5701#
                    5704# 5716  5721# 5735# 5746  5750# 5753# 5765  5770# 5784# 5795  5799# 5802#
                    5814  5819# 5833# 5844  5848# 5851# 5863  5868# 5882# 5901  5906# 5909# 5919
                    5925# 5939# 5950  5954# 5957# 5970  5977# 5992# 6009  6013# 6015# 6039# 6058#
                    6078  6083# 6086# 6099  6104# 6575# 6590  6591# 6614# 6631  6636# 6660# 6677
                    6682# 6706# 6723  6728# 6752# 6769  6774# 6798# 6815  6820# 6844# 6861  6866#
                    6890# 6907  6912# 6936# 6953  6958# 6982# 6999  7004# 7030# 7047  7052# 7078#
                    7095  7100# 7126# 7143  7148# 7173# 7190  7195# 7220# 7237  7242# 7267# 7284
                    7289# 7315# 7332  7337# 7362# 7379  7384# 7411# 7428  7433# 7458# 7475  7480#
                    7505# 7522  7527# 7552# 7569  7574# 7599# 7616  7621# 7646# 7663  7668# 7693#
                    7710  7715# 7740# 7757  7762# 7787# 7804  7809# 7835# 7852  7857# 7882# 7899
                    7904# 7929# 7946  7951# 7976# 7993  7998# 8023# 8040  8045# 8070# 8087  8092#
T$$SOF= 010170      8164# 8167
T$$SUB= 010110      5991# 6044
T$$SW = 010002      3822# 3825
T$$TES= 010166      4737# 4744  4755  4756  4762# 4770  4790# 4822  4830# 4852  4867# 4888  4896#
                    4916  4924# 4944  4951# 4974  4988# 5020  5028# 5061  5069# 5100  5108# 5131
                    5144# 5158  5160  5168# 5210  5218# 5259  5267# 5308  5316# 5357  5365# 5416
                    5424# 5477  5485# 5526  5534# 5575  5583# 5624  5632# 5673  5681# 5722  5730#
                    5771  5779# 5820  5828# 5869  5877# 5926  5934# 5978  5986# 6034  6045  6053#
                    6105  6112# 6131  6138# 6156  6164# 6193  6201# 6231  6238# 6262  6263  6266
                    6273# 6296  6297  6300  6307# 6329  6330  6333  6341# 6383  6395# 6427  6436#
                    6472  6480# 6504  6512# 6539  6553  6557  6568# 6592  6605# 6637  6641  6651#
                    6683  6687  6697# 6729  6733  6743# 6775  6779  6789# 6821  6825  6835# 6867
                    6871  6881# 6913  6917  6927# 6959  6963  6973# 7005  7010  7021# 7053  7057
                    7068# 7101  7106  7117# 7149  7153  7164# 7196  7200  7211# 7243  7247  7258#
                    7290  7294  7306# 7338  7342  7353# 7385  7389  7402# 7434  7438  7449# 7481
                    7485  7496# 7528  7532  7543# 7575  7579  7590# 7622  7626  7637# 7669  7673
                    7684# 7716  7720  7731# 7763  7767  7778# 7810  7814  7826# 7858  7862  7873#
```

			7905	7909	7920#	7952	7956	7967#	7999	8003	8013#	8046	8050	8061#	8093		
			8097														
T1	011366	G	3770	4737#													
T10	013054	G	3770	5028#													
T11	013240	G	3770	5069#													
T12	013426	G	3770	5108#													
T13	013576	G	3770	5144#													
T14	013704	G	3770	5168#													
T15	014134	G	3770	5218#													
T16	014364	G	3770	5267#													
T17	014614	G	3770	5316#													
T18	015044	G	3770	5365#													
T19	015340	G	3770	5424#													
T2	011514	G	3770	4762#													
T20	015634	G	3770	5485#													
T21	016064	G	3770	5534#													
T22	016314	G	3770	5583#													
T23	016544	G	3770	5632#													
T24	016774	G	3770	5681#													
T25	017224	G	3770	5730#													
T26	017454	G	3770	5779#													
T27	017704	G	3770	5828#													
T28	020134	G	3770	5877#													
T29	020432	G	3770	5934#													
T3	011560	G	3770	4790#													
T30	020662	G	3770	5986#													
T30.1	020700		5991#														
T31	021226	G	3770	6053#													
T32	021540	G	3770	6112#													
T33	021702	G	3770	6138#													
T34	022044	G	3770	6164#													
T35	022220	G	3770	6201#													
T36	022424	G	3770	6238#													
T37	022570	G	3770	6273#													
T38	022740	G	3770	6307#													
T39	023104	G	3770	6341#													
T4	011746	G	3770	4830#													
T40	023272	G	3770	6395#													
T41	023506	G	3770	6436#													
T42	023724	G	3770	6480#													
T43	024056	G	3770	6512#													
T44	024274	G	3770	6568#													
T45	024434	G	3770	6605#													
T46	024640	G	3770	6651#													
T47	025044	G	3770	6697#													
T48	025250	G	3770	6743#													
T49	025454	G	3770	6789#													
T5	012112	G	3770	4867#													
T50	025660	G	3770	6835#													
T51	026064	G	3770	6881#													
T52	026270	G	3770	6927#													
T53	026474	G	3770	6973#													
T54	026702	G	3770	7021#													
T55	027106	G	3770	7068#													
T56	027312	G	3770	7117#													
T57	027516	G	3770	7164#													

```
T58      027722 G          3770    7211#
T59      030126 G          3770    7258#
T6       012246 G          3770    4896#
T60      030332 G          3770    7306#
T61      030536 G          3770    7353#
T62      030742 G          3770    7402#
T63      031146 G          3770    7449#
T64      031352 G          3770    7496#
T65      031556 G          3770    7543#
T66      031762 G          3770    7590#
T67      032166 G          3770    7637#
T68      032372 G          3770    7684#
T69      032576 G          3770    7731#
T7       012376 G          3770    4924#
T70      033002 G          3770    7778#
T71      033206 G          3770    7826#
T72      033412 G          3770    7873#
T73      033616 G          3770    7920#
T74      034022 G          3770    7967#
T75      034226 G          3770    8013#
T76      034432 G          3770    8061#
T8       012526 G          3770    4951#
T9       012670 G          3770    4988#
UAM   =  000200 G          3852#
VECTOR   035022            8125    8137#
WMP      034704            8123    8134#
WTYPE    002626            3925#   4577*   4641    4643    4648
X$ALWA=  000000            3706#
X$FALS=  000040            3706#
X$OFFS=  000400            3706#
X$TRUE=  000020            3706#
ZERO     002600            3913#
$BDADR   002634            3928#   4282*   4751*   4768*   4801*   4815*   4841*   4847*   4876*   4884*   4905*   4913*   4933*
                           4941*   4963*   4969*   4999*   5013*   5039*   5054*   5079*   5093*   5119*   5128*   5156*   6126*
                           6128*   6151*   6153*   6190*   6223*   6227*   6261*   6295*   6328*   6381*   6422*   6459*   6469*
                           6502*   6538*   6589*   6630*   6676*   6722*   6768*   6814*   6860*   6906*   6952*   6998*   7046*
                           7094*   7142*   7189*   7236*   7283*   7331*   7378*   7427*   7474*   7521*   7568*   7615*   7662*
                           7709*   7756*   7803*   7851*   7898*   7945*   7992*   8039*   8086*
$BDDAT   002640            3930#   4282*   4455    4456    4457    4458    4459    4460    4461    4462    4463    4464    4467
                           4469    4472    4475    4477    4478    4479    4480    4481    4482    4485    4486    4487    4488
                           4751*   4768*   4801*   4815*   4841*   4847*   4876*   4884*   4905*   4913*   4933*   4941*   4963*
                           4969*   4999*   5013*   5039*   5054*   5079*   5093*   5119*   5128*   5156*   5184*   5203*   5233*
                           5252*   5282*   5301*   5331*   5350*   5384*   5408*   5443*   5469*   5500*   5519*   5549*   5568*
                           5598*   5617*   5647*   5666*   5696*   5715*   5745*   5764*   5794*   5813*   5843*   5862*   5900*
                           5918*   5949*   5969*   6008*   6033*   6077*   6098*   6126*   6128*   6151*   6153*   6190*   6223*
                           6227*   6261*   6295*   6328*   6381*   6422*   6459*   6469*   6502*   6538*   6589*   6630*   6676*
                           6722*   6768*   6814*   6860*   6906*   6952*   6998*   7046*   7094*   7142*   7189*   7236*   7283*
                           7331*   7378*   7427*   7474*   7521*   7568*   7615*   7662*   7709*   7756*   7803*   7851*   7898*
                           7945*   7992*   8039*   8086*
$GDADR   002632            3927#   4458    4459    4460    4461    4462    4492    4668*   4750*   6008*   6033*   6077*   6098*
$GDDAT   002636            3929#   4278*   4280    4282*   4455    4456    4457    4458    4459    4460    4461    4462    4463
                           4464    4467    4469    4472    4475    4477    4478    4479    4480    4481    4482    4485    4486
                           4487    4488    4751*   4768*   4801*   4815*   4836*   4837    4839    4844*   4847*   4876*   4884*
                           4905*   4913*   4933*   4941*   4958*   4959    4961    4966*   4999*   5013*   5039*   5054*   5079*
                           5093*   5119*   5128*   5156*   5184*   5203*   5233*   5252*   5282*   5301*   5331*   5350*   5384*
                           5408*   5443*   5469*   5500*   5519*   5549*   5568*   5598*   5617*   5647*   5666*   5696*   5715*
```

```
                    5745* 5764* 5794* 5813* 5843* 5862* 5900* 5918* 5949* 5969* 6004* 6006  6008*
                    6029* 6031  6033* 6073* 6075  6077* 6094* 6096  6098* 6126* 6128* 6151* 6153*
                    6175* 6190* 6223* 6227* 6257* 6259  6287* 6293  6324* 6326  6359* 6379  6418*
                    6419* 6420  6454* 6457  6465* 6497* 6500  6534* 6536  6585* 6587  6626* 6628
                    6672* 6674  6718* 6720  6764* 6766  6810* 6812  6856* 6858  6902* 6904  6948*
                    6950  6994* 6996  7042* 7044  7090* 7092  7138* 7140  7185* 7187  7232* 7234
                    7279* 7281  7327* 7329  7374* 7376  7423* 7425  7470* 7472  7517* 7519  7564*
                    7566  7611* 7613  7658* 7660  7705* 7707  7752* 7754  7799* 7801  7847* 7849
                    7894* 7896  7941* 7943  7988* 7990  8035* 8037  8082* 8084
$LSTIN= 000000      3715#
$LSTTA= 000000      3716#
$TMP0   002550      3901# 4256* 4259  4274
.       = 040004    3698# 3894# 3896# 3918# 3919# 3920# 3921# 3948# 3994# 4015# 4111# 4514  4744
                    4755  4802  4816  4842  4848  4877  4885  4906  4934  4964  4970  5000  5014
                    5040  5055  5080  5094  5120  5129  5158  5185  5204  5234  5253  5283  5302
                    5332  5351  5385  5409  5445  5470  5501  5520  5550  5569  5599  5618  5648
                    5667  5697  5716  5746  5765  5795  5814  5844  5863  5901  5919  5950  5970
                    6009  6034  6078  6099  6262  6263  6296  6297  6329  6330  6522  6539  6553
                    6590  6631  6637  6677  6683  6723  6729  6769  6775  6815  6821  6861  6867
                    6907  6913  6953  6959  6999  7005  7047  7053  7095  7101  7143  7149  7190
                    7196  7237  7243  7284  7290  7332  7338  7379  7385  7428  7434  7475  7481
                    7522  7528  7569  7575  7616  7622  7663  7669  7710  7716  7757  7763  7804
                    7810  7852  7858  7899  7905  7946  7952  7993  7999  8040  8046  8087  8093
                    8181#
.MSTCL  003156      4105# 4793  4831  4868  4897  4925  4952  4990  5030  5071  5110  5146  5169
                    5219  5268  5317  5366  5425  5486  5535  5584  5633  5682  5731  5780  5829
                    5878  5936  5988  6054  6140  6166  6192  6203  6230  6275  6309  6342  6397
                    6438  6482  6514  6570  6607  6653  6699  6745  6791  6837  6883  6929  6975
                    7023  7071  7119  7166  7213  7260  7308  7355  7404  7451  7498  7545  7592
                    7639  7686  7733  7780  7826  7875  7922  7969  8016  8063
.ROMCL  003244      4120# 4129  4131  4133  4136  4146  4153  4160  4168  4182  4243  4261  4263
                    4266  4271  4275  4302  4313  4315  4323  4325  5177  5179  5196  5198  5226
                    5228  5245  5247  5275  5277  5294  5296  5324  5326  5343  5345  5374  5376
                    5398  5400  5433  5435  5458  5460  5493  5495  5512  5514  5542  5544  5561
                    5563  5591  5593  5610  5612  5640  5642  5659  5661  5689  5691  5708  5710
                    5738  5740  5757  5759  5787  5789  5806  5808  5836  5838  5855  5857  5887
                    5895  5913  5942  5944  5961  5963  5996  6000  6002  6021  6025  6027  6063
                    6068  6071  6090  6092  6122  6147  6181  6219  6251  6254  6284  6288  6290
                    6318  6321  6352  6374  6376  6413  6415  6446  6449  6452  6463  6489  6492
                    6495  6531  6577  6579  6581  6583  6618  6622  6624  6664  6668  6670  6710
                    6714  6716  6756  6760  6762  6802  6806  6808  6848  6852  6854  6894  6898
                    6900  6940  6944  6946  6986  6990  6992  7034  7038  7040  7082  7086  7088
                    7130  7134  7136  7177  7181  7183  7224  7228  7230  7271  7275  7277  7319
                    7323  7325  7366  7370  7372  7415  7419  7421  7462  7466  7468  7509  7513
                    7515  7556  7560  7562  7603  7607  7609  7650  7654  7656  7697  7701  7703
                    7744  7748  7750  7791  7795  7797  7839  7843  7845  7886  7890  7892  7933
                    7937  7939  7980  7984  7986  8027  8031  8033  8074  8078  8080
```

```
BADHEA  4073#  4732  4735  4758  4760  4772  4788  4824  4828  4861  4865  4890  4894  4918  4922
        4946   4949  4981  4985  5022  5026  5063  5067  5102  5106  5136  5142  5162  5166  5212
        5216   5261  5265  5310  5314  5359  5363  5418  5422  5479  5483  5528  5532  5577  5581
        5626   5630  5675  5679  5724  5728  5773  5777  5822  5826  5871  5875  5928  5932  5980
        5984   6047  6051  6107  6110  6133  6136  6158  6162  6195  6199  6233  6236  6268  6271
        6302   6305  6335  6339  6385  6393  6429  6434  6474  6478  6506  6510  6563  6566  6597
        6603   6643  6649  6689  6695  6735  6741  6781  6787  6827  6833  6873  6879  6919  6925
        6965   6971  7013  7019  7060  7066  7109  7115  7156  7162  7203  7209  7250  7256  7297
        7304   7345  7351  7392  7398  7441  7447  7488  7494  7535  7541  7582  7589  7629  7635
        7676   7682  7723  7729  7770  7776  7817  7823  7865  7871  7912  7918  7959  7965  8005
        8011   8053  8059
BCOMPL    15#  3706#  4552  4555  4558
BERROR    19#  3706#  4061#  5184  5203  5233  5252  5282  5301  5331  5350  5384  5408  5443  5469
        5500   5519  5549  5568  5598  5617  5647  5666  5696  5715  5745  5764  5794  5813  5843
        5862   5900  5918  5949  5969
BGNAU     23#  3706#  4718
BGNAUT    31#  3706#  4655
BGNCLN    39#  3706#  4683
BGNDU     47#  3706#  4700
BGNHRD    55#  3706#  8121
BGNHW     66#  3706#  3793
BGNINI    77#  3706#  4537
BGNMOD    85#  3706#  3712
BGNMSG    98#  3706#  4455  4456  4457  4458  4459  4460  4461  4462  4463  4464  4465  4467  4469
        4470   4471  4472  4473  4474  4475  4476  4477  4478  4479  4480  4481  4482  4483  4484
        4485   4486  4487  4488  4490
BGNPRO   106#  3706#  3756
BGNPTA   114#  3706#
BGNRPT   144#  3706#  4506
BGNSEG   152#  3706#  4795  4810  4834  4869  4879  4898  4908  4926  4936  4955  4993  5006  5033
        5046   5073  5088  5112  5123  5174  5192  5223  5241  5272  5290  5321  5339  5370  5393
        5429   5453  5490  5508  5539  5557  5588  5606  5637  5655  5686  5704  5735  5753  5784
        5802   5833  5851  5882  5909  5939  5957  5992  6015  6058  6086  6575  6614  6660  6706
        6752   6798  6844  6890  6936  6982  7030  7078  7126  7173  7220  7267  7315  7362  7411
        7458   7505  7552  7599  7646  7693  7740  7787  7835  7882  7929  7976  8023  8070
BGNSET   161#  3706#
BGNSFT   182#  3706#  8164
BGNSRV   193#  3706#
BGNSUB   201#  3706#  5991
BGNSW    225#  3706#  3822
BGNTST   236#  3706#  4737  4762  4790  4830  4867  4896  4924  4951  4988  5028  5069  5108  5144
        5168   5218  5267  5316  5365  5424  5485  5534  5583  5632  5681  5730  5779  5828  5877
        5934   5986  6053  6112  6138  6164  6201  6238  6273  6307  6341  6395  6436  6480  6512
        6568   6605  6651  6697  6743  6789  6855  6881  6927  6973  7021  7068  7117  7164  7211
        7258   7306  7353  7402  7449  7496  7543  7590  7637  7684  7731  7778  7826  7873  7920
        7967   8013  8061
BNCOMP   266#  3706#  4562  4575
BNERRO   270#  3706#
BREAK    274#  3706#
BRESET   278#  3706#  4684  4702  6114  6189  6239
CKLOOP   282#  3706#
CLOCK    286#  3706#
CLOSE    292#  3706#
CLRVEC   296#  3706#
COMMEN   301#  3706#
DELAY    322#  3706#
```

```
DESCRI     3?7#  3706#  3882
DEVTYP     341#  3706#  4015
DISPAT     346#  3706#  3770
DISPLA     360#  3706#
DOCLN      376#  3706#
DODU       380#  3706#  4669
DORPT      385#  3706#
ED$CAL    4068#  4732   4735  4758  4760  4772  4788  4824  4828  4861  4865  4890  4894  4918  4922
           4946  4949   4981  4985  5022  5026  5063  5067  5102  5106  5136  5142  5162  5166  5212
           5216  5261   5265  5310  5314  5359  5363  5418  5422  5479  5483  5528  5532  5577  5581
           5626  5630   5675  5679  5724  5728  5773  5777  5822  5826  5871  5875  5928  5932  5980
           5984  6047   6051  6107  6110  6133  6136  6158  6162  6195  6199  6233  6236  6268  6271
           6302  6305   6335  6339  6385  6393  6429  6434  6474  6478  6506  6510  6563  6566  6597
           6603  6643   6649  6689  6695  6735  6741  6781  6787  6827  6833  6873  6879  6919  6925
           6965  6971   7013  7019  7060  7066  7109  7115  7156  7162  7203  7209  7250  7256  7297
           7304  7345   7351  7392  7398  7441  7447  7488  7494  7535  7541  7582  7589  7629  7635
           7676  7682   7723  7729  7770  7776  7817  7823  7865  7871  7912  7958  7959  7965  8005
           8011  8053   8059
ENDAU      389#  3706#  4719
ENDAUT     401#  3706#  4673
ENDCLN     413#  3706#  4686
ENDCOM     425#  3706#
ENDDU      441#  3706#  4703
ENDHRD     453#  3706#  8132
ENDHW      465#  3706#  3808
ENDINI     475#  3706#  4652
ENDMCD     487#  3706#  8175
ENDMSG     500#  3706#  4455  4456  4457  4458  4459  4460  4461  4462  4463  4464  4465  4467  4469
           4470  4471   4472  4473  4474  4475  4476  4477  4478  4479  4480  4481  4482  4483  4484
           4485  4486   4487  4488  4493
ENDPRO     512#  3706#  3760
ENDPTA     520#  3706#
ENDRPT     529#  3706#  4523
ENDSEG     541#  3706#  4807  4821  4851  4878  4886  4907  4914  4935  4942  4973  5004  5019  5044
           5060  5087   5099  5121  5130  5189  5209  5238  5258  5287  5307  5336  5356  5390  5415
           5450  5476   5505  5525  5554  5574  5603  5623  5652  5672  5701  5721  5750  5770  5799
           5819  5848   5868  5906  5925  5954  5977  6013  6039  6083  6104  6591  6636  6682  6728
           6774  6820   6866  6912  6958  7004  7052  7100  7148  7195  7242  7289  7337  7384  7433
           7480  7527   7574  7621  7668  7715  7762  7809  7857  7904  7951  7998  8045  8092
ENDSET     555#  3706#
ENDSFT     568#  3706#  8167
ENDSRV     580#  3706#
ENDSUB     596#  3706#  6044
ENDSW      614#  3706#  3825
ENDTST     624#  3706#  4756  4770  4822  4852  4888  4916  4944  4974  5020  5061  5100  5131  5160
           5210  5259   5308  5357  5416  5477  5526  5575  5624  5673  5722  5771  5820  5869  5926
           5978  6045   6105  6131  6156  6193  6231  6266  6300  6333  6383  6427  6472  6504  6557
           6592  6641   6687  6733  6779  6825  6871  6917  6963  7010  7057  7106  7153  7200  7247
           7294  7342   7389  7438  7485  7532  7579  7626  7673  7720  7767  7814  7862  7909  7956
           8003  8050   8097
EQUAL      642#  3706#  3852
ERRDF      714#  3706#  4282  4751  4768  4801  4815  4841  4847  4876  4884  4905  4913  4933  4941
           4963  4969   4999  5013  5039  5054  5079  5093  5119  5128  5156  5184  5203  5233  5252
           5282  5301   5331  5350  5384  5408  5443  5469  5500  5519  5549  5568  5598  5617  5647
           5666  5696   5715  5745  5764  5794  5813  5843  5862  5900  5918  5949  5969  6008  6033
           6077  6098   6126  6128  6151  6153  6190  6223  6227  6261  6295  6328  6381  6422  6459
```

```
                6469    6502    6538    6589    6630    6676    6722    6768    6814    6860    6906    6952    6998    7046    7094
                7142    7189    7236    7283    7331    7378    7427    7474    7521    7568    7615    7662    7709    7756    7803
                7851    7898    7945    7992    8039    8086
ERRHRD           718#    3706#
ERROR            722#    3706#   4046#   4282    4751    4768    4801    4815    4841    4847    4876    4884    4905    4913    4933
                4941    4963    4969    4999    5013    5039    5054    5079    5093    5119    5128    5156    6126    6128    6151
                6153    6190    6223    6227    6261    6295    6328    6381    6422    6459    6469    6502    6538    6589    6630
                6676    6722    6768    6814    6860    6906    6952    6998    7046    7094    7142    7189    7236    7283    7331
                7378    7427    7474    7521    7568    7615    7662    7709    7756    7803    7851    7898    7945    7992    8039
                8086
ERRSF            726#    3706#
ERRSOF           730#    3706#
ERRTBL           734#    3706#
ESCAPE           744#    3706#   4744    4755    4802    4816    4842    4848    4877    4885    4906    4934    4964    4970    5000
                5014    5040    5055    5080    5094    5120    5129    5158    5185    5204    5234    5253    5283    5302    5332
                5351    5385    5409    5445    5470    5501    5520    5550    5569    5599    5618    5648    5667    5697    5716
                5746    5765    5795    5814    5844    5863    5901    5919    5950    5970    6009    6034    6078    6099    6262
                6296    6529    6539    6590    6631    6677    6723    6769    6815    6861    6907    6953    6999    7047    7095
                7143    7190    7237    7284    7332    7379    7428    7475    7522    7569    7616    7663    7710    7757    7804
                7852    7899    7946    7993    8040    8087
EXIT             771#    3706#   4514    6263    6297    6330    6553    6637    6683    6729    6775    6821    6867    6913    6959
                7005    7053    7101    7149    7196    7243    7290    7338    7385    7434    7481    7528    7575    7622    7669
                7716    7763    7810    7858    7905    7952    7999    8046    8093
FEQUAL           810#    3706#
GETBYT           824#    3706#
GETPRI           834#    3706#
GETWOR           829#    3706#
GMANIA           839#    3706#
GMANID           848#    3706#
GMANIL           859#    3706#
GPHARD           868#    3706#   4574
GPRMA            874#    3706#   8124    8125
GPRMD            903#    3706#   8123    8126
GPRML            934#    3706#
HEADER           954#    3706#   3747
INLOOP           962#    3706#
IOSETU           966#    3706#
IOSTAR           974#    3706#
KT11             982#    3706#
K4ONLY          4078#
LASTAD          1147#    3706#   8183
MANUAL          1162#    3706#
MDT0            4419#    4465    4470    4471    4473    4474    4476    4483    4484
MDT1            4422#    4458    4459    4460    4461    4462
MDT2            4426#    4455    4456    4457    4463    4464    4467    4472    4477    4478    4480    4482    4485    4486    4487
                4488
MDT27           4435#    4479    4481
MDT5            4431#    4469    4475
MEMORY          1166#    3706#
MSTCLR          4099#    4793    4831    4868    4897    4925    4952    4990    5030    5071    5110    5146    5169    5219    5268
                5317    5366    5425    5486    5535    5584    5633    5682    5731    5780    5829    5878    5936    5988    6054
                6140    6166    6192    6203    6230    6275    6309    6342    6397    6438    6482    6514    6570    6607    6653
                6699    6745    6791    6837    6883    6929    6975    7023    7071    7119    7166    7213    7260    7308    7355
                7404    7451    7498    7545    7592    7639    7686    7733    7780    7828    7875    7922    7969    8016    8063
MYINT           4087#    4763    4791    5070    5111    5145    5173    5222    5271    5320    5369    5428    5489    5538    5587
                5636    5685    5734    5783    5832    5881    5938    5987    6057    6113    6139    6165    6202    6241    6274
```

	6308	6343	6396	6437	6481	6513	6569	6606	6652	6698	6744	6790	6836	6882	6928
	6974	7022	7069	7118	7165	7212	7259	7307	7354	7403	7450	7497	7544	7591	7638
	7685	7732	7779	7827	7874	7921	7968	8015	8062						
MSBYTE	2000#	3706#	3747#												
MSCHEC	2118#	3706#	4514#	6263#	6297#	6330#	6553#	6637#	6683#	6729#	6775#	6821#	6867#	6913#	6959#
	7005#	7053#	7101#	7149#	7196#	7243#	7290#	7338#	7385#	7434#	7481#	7528#	7575#	7622#	7669#
	7716#	7763#	7810#	7858#	7905#	7952#	7999#	8046#	8093#						
MSCNTO	2182#	3706#	8123#	8124#	8125#	8126#									
MSCOUN	2066#	3706#	4455#	4456#	4457#	4458#	4459#	4460#	4461#	4462#	4463#	4464#	4465#	4467#	4469#
	4470#	4471#	4472#	4473#	4474#	4475#	4476#	4477#	4478#	4479#	4480#	4481#	4482#	4483#	4484#
	4485#	4486#	4487#	4488#	4491#	4492#									
MSDATA	1867#	3706#	3747#	3882#	4015#										
MSDECR	2029#	3706#	3760#	3808#	3825#	4455#	4456#	4457#	4458#	4459#	4460#	4461#	4462#	4463#	4464#
	4465#	4467#	4469#	4470#	4471#	4472#	4473#	4474#	4475#	4476#	4477#	4478#	4479#	4480#	4481#
	4482#	4483#	4484#	4485#	4486#	4487#	4488#	4493#	4523#	4652#	4673#	4686#	4703#	4719#	4756#
	4770#	4807#	4821#	4822#	4851#	4852#	4878#	4886#	4888#	4907#	4914#	4916#	4935#	4942#	4944#
	4973#	4974#	5004#	5019#	5020#	5044#	5060#	5061#	5087#	5099#	5100#	5121#	5130#	5131#	5160#
	5189#	5209#	5210#	5238#	5258#	5259#	5287#	5307#	5308#	5336#	5356#	5357#	5390#	5415#	5416#
	5450#	5476#	5477#	5505#	5525#	5526#	5554#	5574#	5575#	5603#	5623#	5624#	5652#	5672#	5673#
	5701#	5721#	5722#	5750#	5770#	5771#	5799#	5819#	5820#	5848#	5868#	5869#	5906#	5925#	5926#
	5954#	5977#	5978#	6013#	6039#	6044#	6045#	6083#	6104#	6105#	6131#	6156#	6193#	6231#	6266#
	6300#	6333#	6383#	6427#	6472#	6504#	6557#	6591#	6592#	6636#	6641#	6682#	6687#	6728#	6733#
	6774#	6779#	6820#	6825#	6866#	6871#	6912#	6917#	6958#	6963#	7004#	7010#	7052#	7057#	7100#
	7106#	7148#	7153#	7195#	7200#	7242#	7247#	7289#	7294#	7337#	7342#	7384#	7389#	7433#	7438#
	7480#	7485#	7527#	7532#	7574#	7579#	7621#	7626#	7668#	7673#	7715#	7720#	7762#	7767#	7809#
	7814#	7857#	7862#	7904#	7909#	7951#	7956#	7998#	8003#	8045#	8050#	8092#	8097#	8132#	8167#
	8175#														
MSDEFA	2170#	3706#	8123#	8124#	8125#	8126#									
MSENDE	2074#	3706#	3808#	3825#	4455#	4456#	4457#	4458#	4459#	4460#	4461#	4462#	4463#	4464#	4465#
	4467#	4469#	4470#	4471#	4472#	4473#	4474#	4475#	4476#	4477#	4478#	4479#	4480#	4481#	4482#
	4483#	4484#	4485#	4486#	4487#	4488#	4493#	4523#	4652#	4673#	4686#	4703#	4719#	4756#	4770#
	4807#	4821#	4822#	4851#	4852#	4878#	4886#	4888#	4907#	4914#	4916#	4935#	4942#	4944#	4973#
	4974#	5004#	5019#	5020#	5044#	5060#	5061#	5087#	5099#	5100#	5121#	5130#	5131#	5160#	5189#
	5209#	5210#	5238#	5258#	5259#	5287#	5307#	5308#	5336#	5356#	5357#	5390#	5415#	5416#	5450#
	5476#	5477#	5505#	5525#	5526#	5554#	5574#	5575#	5603#	5623#	5624#	5652#	5672#	5673#	5701#
	5721#	5722#	5750#	5770#	5771#	5799#	5819#	5820#	5848#	5868#	5869#	5906#	5925#	5926#	5954#
	5977#	5978#	6013#	6039#	6044#	6045#	6083#	6104#	6105#	6131#	6156#	6193#	6231#	6266#	6300#
	6333#	6383#	6427#	6472#	6504#	6557#	6591#	6592#	6636#	6641#	6682#	6687#	6728#	6733#	6774#
	6779#	6820#	6825#	6866#	6871#	6912#	6917#	6958#	6963#	7004#	7010#	7052#	7057#	7100#	7106#
	7148#	7153#	7195#	7200#	7242#	7247#	7289#	7294#	7337#	7342#	7384#	7389#	7433#	7438#	7480#
	7485#	7527#	7532#	7574#	7579#	7621#	7626#	7668#	7673#	7715#	7720#	7762#	7767#	7809#	7814#
	7857#	7862#	7904#	7909#	7951#	7956#	7998#	8003#	8045#	8050#	8092#	8097#	8132#	8167#	8175#
MSERRI	1649#	3706#	4282#	4751#	4768#	4801#	4815#	4841#	4847#	4876#	4884#	4905#	4913#	4933#	4941#
	4963#	4969#	4999#	5013#	5039#	5054#	5079#	5093#	5119#	5128#	5156#	5184#	5203#	5233#	5252#
	5282#	5301#	5331#	5350#	5384#	5408#	5443#	5469#	5500#	5519#	5549#	5568#	5598#	5617#	5647#
	5666#	5696#	5715#	5745#	5764#	5794#	5813#	5843#	5862#	5900#	5918#	5949#	5969#	6008#	6033#
	6077#	6098#	6126#	6128#	6151#	6153#	6190#	6223#	6227#	6261#	6295#	6328#	6381#	6422#	6459#
	6469#	6502#	6538#	6589#	6630#	6676#	6722#	6768#	6814#	6860#	6906#	6952#	6998#	7046#	7094#
	7142#	7189#	7236#	7283#	7331#	7378#	7427#	7474#	7521#	7568#	7615#	7662#	7709#	7756#	7803#
	7851#	7898#	7945#	7992#	8039#	8086#									
MSESCA	2006#	3706#	4744#	4755#	4802#	4816#	4842#	4848#	4877#	4885#	4906#	4934#	4964#	4970#	5000#
	5014#	5040#	5055#	5080#	5094#	5120#	5129#	5158#	5185#	5204#	5234#	5253#	5283#	5302#	5332#
	5351#	5385#	5409#	5445#	5470#	5501#	5520#	5550#	5569#	5599#	5618#	5648#	5667#	5697#	5716#
	5746#	5765#	5795#	5814#	5844#	5863#	5901#	5919#	5950#	5970#	6009#	6034#	6078#	6099#	6262#
	6296#	6329#	6539#	6590#	6631#	6677#	6723#	6769#	6815#	6861#	6907#	6953#	6999#	7047#	7095#
	7143#	7190#	7237#	7284#	7332#	7379#	7428#	7475#	7522#	7569#	7616#	7663#	7710#	7757#	7804#

	7852#	7899#	7946#	7993#	8040#	8087#									
M$ESCS	2010#	3706#	4744#	4755#	4802#	4816#	4842#	4848#	4877#	4885#	4906#	4934#	4964#	4970#	5000#
	5014#	5040#	5055#	5080#	5094#	5120#	5129#	5158#	5185#	5204#	5234#	5253#	5283#	5302#	5332#
	5351#	5385#	5409#	5445#	5470#	5501#	5520#	5550#	5569#	5599#	5618#	5648#	5667#	5697#	5716#
	5746#	5765#	5795#	5814#	5844#	5863#	5901#	5919#	5950#	5970#	6009#	6034#	6078#	6099#	6262#
	6296#	6329#	6539#	6590#	6631#	6677#	6723#	6769#	6815#	6861#	6907#	6953#	6999#	7047#	7095#
	7143#	7190#	7237#	7284#	7332#	7379#	7428#	7475#	7522#	7569#	7616#	7663#	7710#	7757#	7804#
	7852#	7899#	7946#	7993#	8040#	8087#									
M$EXCP	2101#	3706#	8123#	8124#	8125#	8126#									
M$EXIT	2014#	3706#	4514#	6263#	6297#	6330#	6553#	6637#	6683#	6729#	6775#	6821#	6867#	6913#	6959#
	7005#	7053#	7101#	7149#	7196#	7243#	7290#	7338#	7385#	7434#	7481#	7528#	7575#	7622#	7669#
	7716#	7763#	7810#	7858#	7905#	7952#	7999#	8046#	8093#						
M$EXSE	2022#	3706#	4514#	6263#	6297#	6330#	6553#	6637#	6683#	6729#	6775#	6821#	6867#	6913#	6959#
	7005#	7053#	7101#	7149#	7196#	7243#	7290#	7338#	7385#	7434#	7481#	7528#	7575#	7622#	7669#
	7716#	7763#	7810#	7858#	7905#	7952#	7999#	8046#	8093#						
M$EXTJ	2018#	3706#	4514#	6263#	6297#	6330#	6553#	6637#	6683#	6729#	6775#	6821#	6867#	6913#	6959#
	7005#	7053#	7101#	7149#	7196#	7243#	7290#	7338#	7385#	7434#	7481#	7528#	7575#	7622#	7669#
	7716#	7763#	7810#	7858#	7905#	7952#	7999#	8046#	8093#						
M$GEN	2038#	3706#	3712#	3747#	3756#	3770#	3793#	3808#	3822#	3825#	3882#	4015#	4455#	4456#	4457#
	4458#	4459#	4460#	4461#	4462#	4463#	4464#	4465#	4467#	4469#	4470#	4471#	4472#	4473#	4474#
	4475#	4476#	4477#	4478#	4479#	4480#	4481#	4482#	4483#	4484#	4485#	4486#	4487#	4488#	4490#
	4493#	4506#	4523#	4537#	4652#	4655#	4673#	4683#	4686#	4700#	4703#	4718#	4719#	4737#	4756#
	4762#	4770#	4790#	4807#	4821#	4822#	4830#	4851#	4852#	4867#	4878#	4886#	4888#	4896#	4907#
	4914#	4916#	4924#	4935#	4942#	4944#	4951#	4973#	4974#	4988#	5004#	5019#	5020#	5028#	5044#
	5060#	5061#	5069#	5087#	5099#	5100#	5108#	5121#	5130#	5131#	5144#	5160#	5168#	5189#	5209#
	5210#	5218#	5238#	5258#	5259#	5267#	5287#	5307#	5308#	5316#	5336#	5356#	5357#	5365#	5390#
	5415#	5416#	5424#	5450#	5476#	5477#	5485#	5505#	5525#	5526#	5534#	5554#	5574#	5575#	5583#
	5603#	5623#	5624#	5632#	5652#	5672#	5673#	5681#	5701#	5721#	5722#	5730#	5750#	5770#	5771#
	5779#	5799#	5819#	5820#	5828#	5848#	5868#	5869#	5877#	5906#	5925#	5926#	5934#	5954#	5977#
	5978#	5986#	5991#	6013#	6039#	6044#	6045#	6053#	6083#	6104#	6105#	6112#	6131#	6138#	6156#
	6164#	6193#	6201#	6231#	6238#	6266#	6273#	6300#	6307#	6333#	6341#	6383#	6395#	6427#	6436#
	6472#	6480#	6504#	6512#	6557#	6568#	6591#	6592#	6605#	6636#	6641#	6651#	6682#	6687#	6697#
	6728#	6733#	6743#	6774#	6779#	6789#	6820#	6825#	6835#	6866#	6871#	6881#	6912#	6917#	6927#
	6958#	6963#	6973#	7004#	7010#	7021#	7052#	7057#	7068#	7100#	7106#	7117#	7148#	7153#	7164#
	7195#	7200#	7211#	7242#	7247#	7258#	7289#	7294#	7306#	7337#	7342#	7353#	7384#	7389#	7402#
	7433#	7438#	7449#	7480#	7485#	7496#	7527#	7532#	7543#	7574#	7579#	7590#	7621#	7626#	7637#
	7668#	7673#	7684#	7715#	7720#	7731#	7762#	7767#	7778#	7809#	7814#	7826#	7857#	7862#	7873#
	7904#	7909#	7920#	7951#	7956#	7967#	7998#	8003#	8013#	8045#	8050#	8061#	8092#	8097#	8121#
	8132#	8164#	8167#	8183#											
M$GENB	1938#	3706#													
M$GETS	2035#	3706#	3760#	3808#	3825#	4455#	4456#	4457#	4458#	4459#	4460#	4461#	4462#	4463#	4464#
	4465#	4467#	4469#	4470#	4471#	4472#	4473#	4474#	4475#	4476#	4477#	4478#	4479#	4480#	4481#
	4482#	4483#	4484#	4485#	4486#	4487#	4488#	4493#	4523#	4652#	4673#	4686#	4703#	4719#	4756#
	4770#	4802#	4807#	4816#	4821#	4822#	4842#	4848#	4851#	4852#	4877#	4878#	4885#	4886#	4888#
	4906#	4907#	4914#	4916#	4934#	4935#	4942#	4944#	4964#	4970#	4973#	4974#	5000#	5004#	5014#
	5012#	5020#	5040#	5044#	5055#	5060#	5061#	5080#	5087#	5094#	5099#	5100#	5120#	5121#	5129#
	5130#	5131#	5160#	5185#	5189#	5204#	5209#	5210#	5234#	5238#	5253#	5258#	5259#	5283#	5287#
	5302#	5307#	5308#	5332#	5336#	5351#	5356#	5357#	5385#	5390#	5409#	5415#	5416#	5445#	5450#
	5470#	54 6#	5477#	5501#	5505#	5520#	5525#	5526#	5550#	5554#	5569#	5574#	5575#	5599#	5603#
	5618#	5623#	5624#	5648#	5652#	5667#	5672#	5673#	5697#	5701#	5716#	5721#	5722#	5746#	5750#
	5765#	5770#	5771#	5795#	5799#	5814#	5819#	5820#	5844#	5848#	5863#	5868#	5869#	5901#	5906#
	5919#	5925#	5926#	5950#	5954#	5970#	5977#	5978#	6009#	6013#	6039#	6044#	6045#	6078#	6083#
	6099#	6104#	6105#	6131#	6156#	6193#	6231#	6266#	6300#	6333#	6383#	6427#	6472#	6504#	6557#
	6590#	6591#	6592#	6631#	6636#	6641#	6677#	6682#	6687#	6723#	6728#	6733#	6769#	6774#	6779#
	6815#	6820#	6825#	6861#	6866#	6871#	6907#	6912#	6917#	6953#	6958#	6963#	6999#	7004#	7010#
	7047#	7052#	7057#	7095#	7100#	7106#	7143#	7148#	7153#	7190#	7195#	7200#	7237#	7242#	7247#

```
                7284#  7289#  7294#  7332#  7337#  7342#  7379#  7384#  7389#  7428#  7433#  7438#  7475#  7480#  7485#
                7522#  7527#  7532#  7569#  7574#  7579#  7616#  7621#  7626#  7663#  7668#  7673#  7710#  7715#  7720#
                7757#  7762#  7767#  7804#  7809#  7814#  7852#  7857#  7862#  7899#  7904#  7909#  7946#  7951#  7956#
                7993#  7998#  8003#  8040#  8045#  8050#  8087#  8092#  8097#  8132#  8167#  8175#
M$GETT          1877#  3706#  4514#  4744#  4755#  4802#  4816#  4842#  4848#  4877#  4885#  4906#  4934#  4964#  4970#
                5000#  5014#  5040#  5055#  5080#  5094#  5120#  5129#  5158#  5185#  5204#  5234#  5253#  5283#  5302#
                5332#  5351#  5385#  5409#  5445#  5470#  5501#  5520#  5550#  5569#  5599#  5618#  5648#  5667#  5697#
                5716#  5746#  5765#  5795#  5814#  5844#  5863#  5901#  5919#  5950#  5970#  6009#  6034#  6078#  6099#
                6262#  6263#  6296#  6297#  6329#  6330#  6539#  6553#  6590#  6631#  6637#  6677#  6683#  6723#  6729#
                6769#  6775#  6815#  6821#  6861#  6867#  6907#  6913#  6953#  6959#  6999#  7005#  7047#  7053#  7095#
                7101#  7143#  7149#  7190#  7196#  7237#  7243#  7284#  7290#  7332#  7338#  7379#  7385#  7428#  7434#
                7475#  7481#  7522#  7528#  7569#  7575#  7616#  7622#  7663#  7669#  7710#  7716#  7757#  7763#  7804#
                7810#  7852#  7858#  7899#  7905#  7946#  7952#  7993#  7999#  8040#  8046#  8087#  8093#
M$GNGB          1902#  3706#  3712#  3747#  3756#  3770#  3793#  3822#  3882#  4015#  4455#  4456#  4457#  4458#  4459#
                4460#  4461#  4462#  4463#  4464#  4465#  4467#  4469#  4470#  4471#  4472#  4473#  4474#  4475#  4476#
                4477#  4478#  4479#  4480#  4481#  4482#  4483#  4484#  4485#  4486#  4487#  4488#  4490#  4506#  4537#
                4655#  4683#  4700#  4718#  8121#  8164#  8183#
M$GNIN          2049#  3706#  3747#  3770#  3793#  3822#  3882#  4015#  4282#  4455#  4456#  4457#  4458#  4459#  4460#
                4461#  4462#  4463#  4464#  4465#  4467#  4469#  4470#  4471#  4472#  4473#  4474#  4475#  4476#  4477#
                4478#  4479#  4480#  4481#  4482#  4483#  4484#  4485#  4486#  4487#  4488#  4491#  4492#  4493#  4514#
                4523#  4551#  4552#  4554#  4555#  4557#  4558#  4561#  4562#  4574#  4575#  4652#  4669#  4673#  4684#
                4686#  4702#  4703#  4719#  4744#  4751#  4755#  4756#  4768#  4770#  4795#  4801#  4802#  4807#  4810#
                4815#  4816#  4821#  4822#  4834#  4841#  4842#  4847#  4848#  4851#  4852#  4869#  4876#  4877#  4878#
                4879#  4884#  4885#  4886#  4888#  4898#  4905#  4906#  4907#  4908#  4913#  4914#  4916#  4926#  4933#
                4934#  4935#  4936#  4941#  4942#  4944#  4955#  4963#  4964#  4969#  4970#  4973#  4974#  4993#  4999#
                5000#  5004#  5006#  5013#  5014#  5019#  5020#  5033#  5039#  5040#  5044#  5046#  5054#  5055#  5060#
                5061#  5073#  5079#  5080#  5087#  5088#  5093#  5094#  5099#  5100#  5112#  5119#  5120#  5121#  5123#
                5128#  5129#  5130#  5131#  5156#  5158#  5160#  5174#  5184#  5185#  5189#  5192#  5203#  5204#  5209#
                5210#  5223#  5233#  5234#  5238#  5241#  5252#  5253#  5258#  5259#  5272#  5282#  5283#  5287#  5290#
                5301#  5302#  5307#  5308#  5321#  5331#  5332#  5336#  5339#  5350#  5351#  5556#  5357#  5370#  5384#
                5385#  5390#  5393#  5408#  5409#  5415#  5416#  5429#  5443#  5445#  5450#  5453#  5469#  5470#  5476#
                5477#  5490#  5500#  5501#  5505#  5508#  5519#  5520#  5525#  5526#  5539#  5549#  5550#  5554#  5557#
                5568#  5569#  5574#  5575#  5588#  5598#  5599#  5603#  5606#  5617#  5618#  5623#  5624#  5637#  5647#
                5648#  5652#  5655#  5666#  5667#  5672#  5673#  5686#  5696#  5697#  5701#  5704#  5715#  5716#  5721#
                5722#  5735#  5745#  5746#  5750#  5753#  5764#  5765#  5770#  5771#  5784#  5794#  5795#  5799#  5802#
                5813#  5814#  5819#  5820#  5833#  5843#  5844#  5848#  5851#  5862#  5863#  5868#  5869#  5882#  5900#
                5901#  5906#  5909#  5918#  5919#  5925#  5926#  5939#  5949#  5950#  5954#  5957#  5969#  5970#  5977#
                5978#  5991#  5992#  6008#  6009#  6013#  6015#  6033#  6034#  6039#  6044#  6045#  6058#  6077#  6078#
                6083#  6086#  6098#  6099#  6104#  6105#  6114#  6120#  6124#  6126#  6128#  6131#  6145#  6149#  6151#
                6155#  6156#  6168#  6183#  6189#  6190#  6193#  6205#  6221#  6223#  6227#  6231#  6239#  6261#  6262#
                6263#  6266#  6295#  6296#  6297#  6300#  6328#  6329#  6330#  6333#  6381#  6383#  6422#  6427#  6459#
                6469#  6472#  6502#  6504#  6538#  6539#  6553#  6557#  6575#  6589#  6590#  6591#  6592#  6614#  6630#
                6631#  6636#  6637#  6641#  6660#  6676#  6677#  6682#  6683#  6687#  6706#  6722#  6723#  6728#  6729#
                6733#  6752#  6768#  6769#  6774#  6775#  6779#  6798#  6814#  6815#  6820#  6821#  6825#  6844#  6860#
                6861#  6866#  6867#  6871#  6890#  6906#  6907A  6912#  6913#  6917#  6936#  6952#  6953#  6958#  6959#
                6963#  6982#  6998#  6999#  7004#  7005#  7010#  7030#  7046#  7047#  7052#  7053#  7057#  7078#  7094#
                7095#  7100#  7101#  7106#  7126#  7142#  7143#  7148#  7149#  7153#  7173#  7189#  7190#  7195#  7196#
                7200#  7220#  7236#  7237#  7242#  7243#  7247#  7267#  7283#  7284#  7289#  7290#  7294#  7315#  7331#
                7332#  7337#  7338#  7342#  7362#  7378#  7379#  7384#  7385#  7389#  7411#  7427#  7428#  7433#  7434#
                7438#  7458#  7474#  7475#  7480#  7481#  7485#  7505#  7521#  7522#  7527#  7528#  7532#  7552#  7568#
                7569#  7574#  7575#  7579#  7599#  7615#  7616#  7621#  7622#  7626#  7646#  7662#  7663#  7668#  7669#
                7673#  7693#  7709#  7710#  7715#  7716#  7720#  7740#  7756#  7757#  7762#  7763#  7767#  7787#  7803#
                7804#  7809#  7810#  7814#  7835#  7851#  7852#  7857#  7858#  7862#  7882#  7898#  7899#  7904#  7905#
                7909#  7929#  7945#  7946#  7951#  7952#  7956#  7976#  7992#  7993#  7998#  7999#  8003#  8023#  8039#
                8040#  8045#  8046#  8050#  8070#  8086#  8087#  8092#  8093#  8097#  8121#  8123#  8124#  8125#  8126#
                8132#  8164#  8167#  8183#
```

M$GNLS 1913# 3706# 4807# 4821# 4851# 4878# 4886# 4907# 4914# 4935# 4942# 4973# 5004# 5019# 5044#
5060# 5087# 5099# 5121# 5130# 5189# 5209# 5238# 5258# 5287# 5307# 5336# 5356# 5390# 5415#
5450# 5476# 5505# 5525# 5554# 5574# 5603# 5623# 5652# 5672# 5701# 5721# 5750# 5770# 5799#
5819# 5848# 5868# 5906# 5925# 5954# 5977# 6013# 6039# 6083# 6104# 6591# 6636# 6682# 6728#
6774# 6820# 6866# 6912# 6958# 7004# 7052# 7100# 7148# 7195# 7242# 7289# 7337# 7384# 7433#
7480# 7527# 7574# 7621# 7668# 7715# 7762# 7809# 7857# 7904# 7951# 7998# 8045# 8092#
M$GNSU 1898# 3706# 5991#
M$GNTA 1890# 3706# 3808# 3825# 4455# 4456# 4457# 4458# 4459# 4460# 4461# 4462# 4463# 4464# 4465#
4467# 4469# 4470# 4471# 4472# 4473# 4474# 4475# 4476# 4477# 4478# 4479# 4480# 4481# 4482#
4483# 4484# 4485# 4486# 4487# 4488# 4493# 4523# 4652# 4673# 4686# 4703# 4719# 4756# 4770#
4822# 4852# 4888# 4916# 4944# 4974# 5020# 5061# 5100# 5131# 5160# 5210# 5259# 5308# 5357#
5416# 5477# 5526# 5575# 5624# 5673# 5722# 5771# 5820# 5869# 5926# 5978# 6044# 6045# 6105#
6131# 6156# 6193# 6231# 6266# 6300# 6333# 6383# 6427# 6472# 6504# 6557# 6592# 6641# 6687#
6733# 6779# 6825# 6871# 6917# 6963# 7010# 7057# 7106# 7153# 7200# 7247# 7294# 7342# 7389#
7438# 7485# 7532# 7579# 7626# 7673# 7720# 7767# 7814# 7862# 7909# 7956# 8003# 8050# 8097#
8132# 8167#
M$GNTE 1894# 3706# 4737# 4762# 4790# 4830# 4867# 4896# 4924# 4951# 4988# 5028# 5069# 5108# 5144#
5168# 5218# 5267# 5316# 5365# 5424# 5485# 5534# 5583# 5632# 5681# 5730# 5779# 5828# 5877#
5934# 5986# 6053# 6112# 6138# 6164# 6201# 6238# 6273# 6307# 6341# 6395# 6436# 6480# 6512#
6568# 6605# 6651# 6697# 6743# 6789# 6835# 6881# 6927# 6973# 7021# 7068# 7117# 7164# 7211#
7258# 7306# 7353# 7402# 7449# 7496# 7543# 7590# 7637# 7684# 7731# 7778# 7826# 7873# 7920#
7967# 8013# 8061#
M$HAPT 1739# 3706# 3747#
M$HNAP 1824# 3706# 3747#
M$INCR 2026# 3706# 3712# 3756# 3793# 3822# 4282# 4455# 4456# 4457# 4458# 4459# 4460# 4461# 4462#
4463# 4464# 4465# 4467# 4469# 4470# 4471# 4472# 4473# 4474# 4475# 4476# 4477# 4478# 4479#
4480# 4481# 4482# 4483# 4484# 4485# 4486# 4487# 4488# 4490# 4491# 4492# 4493# 4506# 4523#
4537# 4551# 4554# 4557# 4561# 4574# 4652# 4655# 4669# 4673# 4683# 4684# 4686# 4700# 4702#
4703# 4718# 4719# 4737# 4744# 4751# 4755# 4756# 4762# 4768# 4770# 4790# 4795# 4801# 4802#
4807# 4810# 4815# 4816# 4821# 4822# 4830# 4834# 4841# 4842# 4847# 4848# 4851# 4852# 4867#
4869# 4876# 4877# 4878# 4879# 4884# 4885# 4886# 4888# 4896# 4898# 4905# 4906# 4907# 4908#
4913# 4914# 4916# 4924# 4926# 4933# 4934# 4935# 4936# 4941# 4942# 4944# 4951# 4955# 4963#
4964# 4969# 4970# 4973# 4974# 4988# 4993# 4999# 5000# 5004# 5006# 5013# 5014# 5019# 5020#
5028# 5033# 5039# 5040# 5044# 5046# 5054# 5055# 5060# 5061# 5069# 5073# 5079# 5080# 5087#
5088# 5093# 5094# 5099# 5100# 5108# 5112# 5119# 5120# 5121# 5123# 5128# 5129# 5130# 5131#
5144# 5156# 5158# 5160# 5168# 5174# 5184# 5185# 5189# 5192# 5203# 5204# 5209# 5210# 5218#
5223# 5233# 5234# 5238# 5241# 5252# 5253# 5258# 5259# 5267# 5272# 5282# 5283# 5287# 5290#
5301# 5302# 5307# 5308# 5316# 5321# 5331# 5332# 5336# 5339# 5350# 5351# 5356# 5357# 5365#
5370# 5384# 5385# 5390# 5393# 5408# 5409# 5415# 5416# 5424# 5429# 5443# 5445# 5450# 5453#
5469# 5470# 5476# 5477# 5485# 5490# 5500# 5501# 5505# 5508# 5519# 5520# 5525# 5526# 5534#
5539# 5549# 5550# 5554# 5557# 5568# 5569# 5574# 5575# 5583# 5588# 5598# 5599# 5603# 5606#
5617# 5618# 5623# 5624# 5632# 5637# 5647# 5648# 5652# 5655# 5666# 5667# 5672# 5673# 5681#
5686# 5696# 5697# 5701# 5704# 5715# 5716# 5721# 5722# 5730# 5735# 5745# 5746# 5750# 5753#
5764# 5765# 5770# 5771# 5779# 5784# 5794# 5795# 5799# 5802# 5813# 5814# 5819# 5820# 5828#
5833# 5843# 5844# 5848# 5851# 5862# 5863# 5868# 5869# 5877# 5882# 5900# 5901# 5906# 5909#
5918# 5919# 5925# 5926# 5934# 5939# 5949# 5950# 5954# 5957# 5969# 5970# 5977# 5978# 5986#
5991# 5992# 6008# 6009# 6013# 6015# 6033# 6034# 6039# 6044# 6045# 6053# 6058# 6077# 6078#
6083# 6086# 6098# 6099# 6104# 6105# 6112# 6114# 6120# 6124# 6126# 6128# 6131# 6138# 6145#
6149# 6151# 6153# 6156# 6164# 6168# 6183# 6189# 6190# 6193# 6201# 6205# 6221# 6223# 6227#
6231# 6238# 6239# 6261# 6262# 6263# 6266# 6273# 6295# 6296# 6297# 6300# 6307# 6328# 6329#
6330# 6333# 6341# 6381# 6383# 6395# 6422# 6427# 6436# 6459# 6469# 6472# 6480# 6502# 6504#
6512# 6538# 6539# 6553# 6557# 6568# 6575# 6589# 6590# 6591# 6592# 6605# 6614# 6630# 6631#
6636# 6637# 6641# 6651# 6660# 6676# 6677# 6682# 6683# 6687# 6697# 6706# 6722# 6723# 6728#
6729# 6733# 6743# 6752# 6768# 6769# 6774# 6775# 6779# 6789# 6798# 6814# 6815# 6820# 6821#
6825# 6835# 6844# 6860# 6861# 6866# 6867# 6871# 6881# 6890# 6906# 6907# 6912# 6913# 6917#
6927# 6936# 6952# 6953# 6958# 6959# 6963# 6973# 6982# 6998# 6999# 7004# 7005# 7010# 7021#

	7030#	7046#	7047#	7052#	7053#	7057#	7068#	7078#	7094#	7095#	7100#	7101#	7106#	7117#	7126#
	7142#	7143#	7148#	7149#	7153#	7164#	7173#	7189#	7190#	7195#	7196#	7200#	7211#	7220#	7236#
	7237#	7242#	7243#	7247#	7258#	7267#	7283#	7284#	7289#	7290#	7294#	7306#	7315#	7331#	7332#
	7337#	7338#	7342#	7353#	7362#	7378#	7379#	7384#	7385#	7389#	7402#	7411#	7427#	7428#	7433#
	7434#	7438#	7449#	7458#	7474#	7475#	7480#	7481#	7485#	7496#	7505#	7521#	7522#	7527#	7528#
	7532#	7543#	7552#	7568#	7569#	7574#	7575#	7579#	7590#	7599#	7615#	7616#	7621#	7622#	7626#
	7637#	7646#	7662#	7663#	7668#	7669#	7673#	7684#	7693#	7709#	7710#	7715#	7716#	7720#	7731#
	7740#	7756#	7757#	7762#	7763#	7767#	7778#	7787#	7803#	7804#	7809#	7810#	7814#	7826#	7835#
	7851#	7852#	7857#	7858#	7862#	7873#	7882#	7898#	7899#	7904#	7905#	7909#	7920#	7929#	7945#
	7946#	7951#	7952#	7956#	7967#	7976#	7992#	7993#	7998#	7999#	8003#	8013#	8023#	8039#	8040#
	8045#	8046#	8050#	8061#	8070#	8086#	8087#	8092#	8093#	8097#	8121#	8164#			
M$IOSE	1700#	3706#													
M$LDRO	1942#	3706#	4551#	4554#	4557#	4561#	4574#	4669#	6120#	6124#	6145#	6149#	6168#	6183#	6205#
	6221#														
M$MASK	1671#	3706#													
M$MCH:	4#	3706#													
M$MCLO	1624#	3706#													
M$MSK1	1677#	3706#													
M$POP	1881#	3706#	3760#	3808#	3825#	4455#	4456#	4457#	4458#	4459#	4460#	4461#	4462#	4463#	4464#
	4465#	4467#	4469#	4470#	4471#	4472#	4473#	4474#	4475#	4476#	4477#	4478#	4479#	4480#	4481#
	4482#	4483#	4484#	4485#	4486#	4487#	4488#	4493#	4523#	4652#	4673#	4686#	4703#	4719#	4756#
	4770#	4807#	4821#	4822#	4851#	4852#	4878#	4886#	4888#	4907#	4914#	4916#	4935#	4942#	4944#
	4973#	4974#	5004#	5019#	5020#	5044#	5060#	5061#	5087#	5099#	5100#	5121#	5130#	5131#	5160#
	5189#	5209#	5210#	5238#	5258#	5259#	5287#	5307#	5308#	5336#	5356#	5357#	5390#	5415#	5416#
	5450#	5476#	5477#	5505#	5525#	5526#	5554#	5574#	5575#	5603#	5623#	5624#	5652#	5672#	5673#
	5701#	5721#	5722#	5750#	5770#	5771#	5799#	5819#	5820#	5848#	5868#	5869#	5906#	5925#	5926#
	5954#	5977#	5978#	6013#	6039#	6044#	6045#	6083#	6104#	6105#	6131#	6156#	6193#	6231#	6266#
	6300#	6333#	6383#	6427#	6472#	6504#	6557#	6591#	6592#	6636#	6641#	6682#	6687#	6728#	6733#
	6774#	6779#	6820#	6825#	6866#	6871#	6912#	6917#	6958#	6963#	7004#	7010#	7052#	7057#	7100#
	7106#	7148#	7153#	7195#	7200#	7242#	7247#	7289#	7294#	7337#	7342#	7384#	7389#	7433#	7438#
	7480#	7485#	7527#	7532#	7574#	7579#	7621#	7626#	7668#	7673#	7715#	7720#	7762#	7767#	7809#
	7814#	7857#	7862#	7904#	7909#	7951#	7956#	7998#	8003#	8045#	8050#	8092#	8097#	8132#	8167#
	8175#														
M$PRIN	1636#	3706#	4455#	4456#	4457#	4458#	4459#	4460#	4461#	4462#	4463#	4464#	4465#	4467#	4469#
	4470#	4471#	4472#	4473#	4474#	4475#	4476#	4477#	4478#	4479#	4480#	4481#	4482#	4483#	4484#
	4485#	4486#	4487#	4488#	4491#	4492#									
M$PUSH	1631#	3706#	3712#	3756#	3793#	3822#	4455#	4456#	4457#	4458#	4459#	4460#	4461#	4462#	4463#
	4464#	4465#	4467#	4469#	4470#	4471#	4472#	4473#	4474#	4475#	4476#	4477#	4478#	4479#	4480#
	4481#	4482#	4483#	4484#	4485#	4486#	4487#	4488#	4490#	4506#	4537#	4655#	4683#	4700#	4718#
	4737#	4762#	4790#	4795#	4810#	4830#	4834#	4867#	4869#	4879#	4896#	4898#	4908#	4924#	4926#
	4936#	4951#	4955#	4988#	4993#	5006#	5028#	5033#	5046#	5069#	5073#	5088#	5108#	5112#	5123#
	5144#	5168#	5174#	5192#	5218#	5223#	5241#	5267#	5272#	5290#	5316#	5321#	5339#	5365#	5370#
	5393#	5424#	5429#	5453#	5485#	5490#	5508#	5534#	5539#	5557#	5583#	5588#	5606#	5632#	5637#
	5655#	5681#	5686#	5704#	5730#	5735#	5753#	5779#	5784#	5802#	5828#	5833#	5851#	5877#	5882#
	5909#	5934#	5939#	5957#	5986#	5991#	5992#	6015#	6053#	6058#	6086#	6112#	6138#	6164#	6201#
	6238#	6273#	6307#	6341#	6395#	6436#	6480#	6512#	6568#	6575#	6605#	6614#	6651#	6660#	6697#
	6706#	6743#	6752#	6789#	6798#	6835#	6844#	6881#	6890#	6927#	6936#	6973#	6982#	7021#	7030#
	7068#	7078#	7117#	7126#	7164#	7173#	7211#	7220#	7258#	7267#	7306#	7315#	7353#	7362#	7402#
	7411#	7449#	7458#	7496#	7505#	7543#	7552#	7590#	7599#	7637#	7646#	7684#	7693#	7731#	7740#
	7778#	7787#	7826#	7835#	7873#	7882#	7920#	7929#	7967#	7976#	8013#	8023#	8061#	8070#	8121#
	8164#														
M$PUT	1972#	3706#	4455#	4456#	4457#	4458#	4459#	4460#	4461#	4462#	4463#	4464#	4465#	4467#	4469#
	4470#	4471#	4472#	4473#	4474#	4475#	4476#	4477#	4478#	4479#	4480#	4481#	4482#	4483#	4484#
	4485#	4486#	4487#	4488#	4491#	4492#									
M$PUT1	1981#	3706#	4455#	4456#	4457#	4458#	4459#	4460#	4461#	4462#	4463#	4464#	4465#	4467#	4469#
	4470#	4471#	4472#	4473#	4474#	4475#	4476#	4477#	4478#	4479#	4480#	4481#	4482#	4483#	4484#

```
        4485#  4486#  4487#  4488#  4491#  4492#
M$RADI  2077#  3706#  8123#  8124#  8125#  8126#
M$RBRO  1952#  3706#
M$RNRO  1962#  3706#  4574#
M$SETS  2032#  3706#  3712#  3756#  3793#  3822#  4455#  4456#  4457#  4458#  4459#  4460#  4461#  4462#  4463#
        4464#  4465#  4467#  4469#  4470#  4471#  4472#  4473#  4474#  4475#  4476#  4477#  4478#  4479#  4480#
        4481#  4482#  4483#  4484#  4485#  4486#  4487#  4488#  4490#  4506#  4537#  4655#  4683#  4700#  4718#
        4737#  4762#  4790#  4795#  4810#  4830#  4834#  4867#  4869#  4879#  4896#  4898#  4908#  4924#  4926#
        4936#  4951#  4955#  4988#  4993#  5006#  5028#  5033#  5046#  5069#  5073#  5088#  5108#  5112#  5123#
        5144#  5168#  5174#  5192#  5218#  5223#  5241#  5267#  5272#  5290#  5316#  5321#  5339#  5365#  5370#
        5393#  5424#  5429#  5453#  5485#  5490#  5508#  5534#  5539#  5557#  5583#  5588#  5606#  5632#  5637#
        5655#  5681#  5686#  5704#  5730#  5735#  5753#  5779#  5784#  5802#  5828#  5833#  5851#  5877#  5882#
        5909#  5934#  5939#  5957#  5986#  5991#  5992#  6015#  6053#  6058#  6086#  6112#  6138#  6164#  6201#
        6238#  6273#  6307#  6341#  6395#  6436#  6480#  6512#  6568#  6575#  6605#  6614#  6651#  6660#  6697#
        6706#  6743#  6752#  6789#  6798#  6835#  6844#  6881#  6890#  6927#  6936#  6973#  6982#  7021#  7030#
        7068#  7078#  7117#  7126#  7164#  7173#  7211#  7220#  7258#  7267#  7306#  7315#  7353#  7362#  7402#
        7411#  7449#  7458#  7496#  7505#  7543#  7552#  7590#  7599#  7637#  7646#  7684#  7693#  7731#  7740#
        7778#  7787#  7826#  7835#  7873#  7882#  7920#  7929#  7967#  7976#  8013#  8023#  8061#  8070#  8121#
        8164#
M$STAR  1733#  3706#
M$SVC   1933#  3706#  4282  4455#  4456#  4457#  4458#  4459#  4460#  4461#  4462#  4463#  4464#  4465#  4467#
        4469#  4470#  4471#  4472#  4473#  4474#  4475#  4476#  4477#  4478#  4479#  4480#  4481#  4482#  4483#
        4484#  4485#  4486#  4487#  4488#  4491#  4492#  4493#  4514#  4523#  4551#  4554#  4557#  4561#  4574#
        4652#  4669#  4673#  4684#  4686#  4702#  4703#  4719#  4744#  4751  4755#  4756#  4768  4770#  4795#
        4801  4802#  4807#  4810#  4815  4816#  4821#  4822#  4834#  4841  4842#  4847  4848#  4851#  4852#
        4869#  4876  4877#  4878#  4879#  4884  4885#  4886#  4888#  4898#  4905  4906#  4907#  4908#  4913
        4914#  4916#  4926#  4933  4934#  4935#  4936#  4941  4942#  4944#  4955#  4963  4964#  4969  4970#
        4973#  4974#  4993#  4999  5000#  5004#  5006#  5013  5014#  5019#  5020#  5033#  5039  5040#  5044#
        5046#  5054  5055#  5060#  5061#  5073#  5079  5080#  5087#  5088#  5093  5094#  5099#  5100#  5112#
        5119  5120#  5121#  5123#  5126  5129#  5130#  5131#  5156  5158#  5160#  5174#  5184  5185#  5189#
        5192#  5203  5204#  5209#  5210#  5223#  5233  5234#  5238#  5241#  5252  5253#  5258#  5259#  5272#
        5282  5283#  5287#  5290#  5301  5302#  5307#  5308#  5321#  5331  5332#  5336#  5339#  5350  5351#
        5356#  5357#  5370#  5384  5385#  5390#  5393#  5408  5409#  5415#  5416#  5429#  5443  5445#  5450#
        5453#  5469  5470#  5476#  5477#  5490#  5500  5501#  5505#  5508#  5519  5520#  5525#  5526#  5539#
        5549  5550#  5554#  5557#  5568  5569#  5574#  5575#  5588#  5598  5599#  5603#  5606#  5617  5618#
        5623#  5624#  5637#  5647  5648#  5652#  5655#  5666  5667#  5672#  5673#  5686#  5696  5697#  5701#
        5704#  5715  5716#  5721#  5722#  5735#  5745  5746#  5750#  5753#  5764  5765#  5770#  5771#  5784#
        5794  5795#  5799#  5802#  5813  5814#  5819#  5820#  5833#  5843  5844#  5848#  5851#  5862  5863#
        5868#  5869#  5882#  5900  5901#  5906#  5909#  5918  5919#  5925#  5926#  5939#  5949  5950#  5954#
        5957#  5969  5970#  5977#  5978#  5991#  5992#  6008  6009#  6013#  6015#  6033  6034#  6039#  6044#
        6045#  6058#  6077  6078#  6083#  6086#  6098  6099#  6104#  6105#  6114#  6120#  6124#  6126  6128
        6131#  6145#  6149#  6151  6153  6156#  6168#  6183#  6189#  6190  6193#  6205#  6221#  6223  6227
        6231#  6239#  6261  6262#  6263#  6266#  6295  6296#  6297#  6300#  6328  6329#  6330#  6333#  6381
        6383#  6422  6427#  6459  6469  6472#  6502  6504#  6538  6539#  6553#  6557#  6575#  6589  6590#
        6591#  6592#  6614#  6630  6631#  6636#  6637#  6641#  6660#  6676  6677#  6682#  6683#  6687#  6706#
        6722  6723#  6728#  6729#  6733#  6752#  6768  6769#  6774#  6775#  6779#  6798#  6814  6815#  6820#
        6821#  6825#  6844#  6860  6861#  6866#  6867#  6871#  6890#  6906  6907#  6912#  6913#  6917#  6936#
        6952  6953#  6958#  6959#  6963#  6982#  6998  6999#  7004#  7005#  7010#  7030#  7046  7047#  7052#
        7053#  7057#  7078#  7094  7095#  7100#  7101#  7106#  7126#  7142  7143#  7148#  7149#  7153#  7173#
        7189  7190#  7195#  7196#  7200#  7220#  7236  7237#  7242#  7243#  7247#  7267#  7283  7284#  7289#
        7290#  7294#  7315#  7331  7332#  7337#  7338#  7342#  7362#  7378  7379#  7384#  7385#  7389#  7411#
        7427  7428#  7433#  7434#  7438#  7458#  7474  7475#  7480#  7481#  7485#  7505#  7521  7522#  7527#
        7528#  7532#  7552#  7568  7569#  7574#  7575#  7579#  7599#  7615  7616#  7621#  7622#  7626#  7646#
        7662  7663#  7668#  7669#  7673#  7693#  7709  7710#  7715#  7716#  7720#  7740#  7756  7757#  7762#
        7763#  7767#  7787#  7803  7804#  7809#  7810#  7814#  7835#  7851  7852#  7857#  7858#  7862#  7882#
        7898  7899#  7904#  7905#  7909#  7929#  7945  7946#  7951#  7952#  7956#  7976#  7992  7993#  7998#
```

	7999#	8003#	8023#	8039	8040#	8045#	8046#	8050#	8070#	8086	8087#	8092#	8093#	8097#	
M$TLAB	1929#	3706#	4282#	4455#	4456#	4457#	4458#	4459#	4460#	4461#	4462#	4463#	4464#	4465#	4467#
	4469#	4470#	4471#	4472#	4473#	4474#	4475#	4476#	4477#	4478#	4479#	4480#	4481#	4482#	4483#
	4484#	4485#	4486#	4487#	4488#	4491#	4492#	4493#	4523#	4551#	4554#	4557#	4561#	4574#	4652#
	4669#	4673#	4684#	4686#	4702#	4703#	4719#	4744#	4751#	4755#	4756#	4768#	4770#	4795#	4801#
	4802#	4807#	4810#	4815#	4816#	4821#	4822#	4834#	4841#	4842#	4847#	4848#	4851#	4852#	4869#
	4876#	4877#	4878#	4879#	4884#	4885#	4886#	4888#	4898#	4905#	4906#	4907#	4908#	4913#	4914#
	4916#	4926#	4933#	4934#	4935#	4936#	4941#	4942#	4944#	4955#	4963#	4964#	4969#	4970#	4973#
	4974#	4993#	4999#	5000#	5004#	5006#	5013#	5014#	5019#	5020#	5033#	5039#	5040#	5044#	5046#
	5054#	5055#	5060#	5061#	5073#	5079#	5080#	5087#	5088#	5093#	5094#	5099#	5100#	5112#	5119#
	5120#	5121#	5123#	5128#	5129#	5130#	5131#	5156#	5158#	5160#	5174#	5184#	5185#	5189#	5192#
	5203#	5204#	5209#	5210#	5223#	5233#	5234#	5238#	5241#	5252#	5253#	5258#	5259#	5272#	5282#
	5283#	5287#	5290#	5301#	5302#	5307#	5308#	5321#	5331#	5332#	5336#	5339#	5350#	5351#	5356#
	5357#	5370#	5384#	5385#	5390#	5393#	5408#	5409#	5415#	5416#	5429#	5443#	5445#	5450#	5453#
	5469#	5470#	5476#	5477#	5490#	5500#	5501#	5505#	5508#	5519#	5520#	5525#	5526#	5539#	5549#
	5550#	5554#	5557#	5568#	5569#	5574#	5575#	5588#	5598#	5599#	5603#	5606#	5617#	5618#	5623#
	5624#	5637#	5647#	5648#	5652#	5655#	5666#	5667#	5672#	5673#	5686#	5696#	5697#	5701#	5704#
	5715#	5716#	5721#	5722#	5735#	5745#	5746#	5750#	5753#	5764#	5765#	5770#	5771#	5784#	5794#
	5795#	5799#	5802#	5813#	5814#	5819#	5820#	5833#	5843#	5844#	5848#	5851#	5862#	5863#	5868#
	5869#	5882#	5900#	5901#	5906#	5909#	5918#	5919#	5925#	5926#	5939#	5949#	5950#	5954#	5957#
	5969#	5970#	5977#	5978#	5991#	5992#	6008#	6009#	6013#	6015#	6033#	6034#	6039#	6044#	6045#
	6058#	6077#	6078#	6083#	6086#	6098#	6099#	6104#	6105#	6114#	6120#	6124#	6126#	6128#	6131#
	6145#	6149#	6151#	6153#	6156#	6168#	6183#	6189#	6190#	6193#	6205#	6221#	6223#	6227#	6231#
	6239#	6261#	6262#	6263#	6266#	6295#	6296#	6297#	6300#	6328#	6329#	6330#	6333#	6381#	6383#
	6422#	6427#	6459#	6469#	6472#	6502#	6504#	6538#	6539#	6553#	6557#	6575#	6589#	6590#	6591#
	6592#	6614#	6630#	6631#	6636#	6637#	6641#	6660#	6676#	6677#	6682#	6683#	6687#	6706#	6722#
	6723#	6728#	6729#	6733#	6752#	6768#	6769#	6774#	6775#	6779#	6798#	6814#	6815#	6820#	6821#
	6825#	6844#	6860#	6861#	6866#	6867#	6871#	6890#	6906#	6907#	6912#	6913#	6917#	6936#	6952#
	6953#	6958#	6959#	6963#	6982#	6998#	6999#	7004#	7005#	7010#	7030#	7046#	7047#	7052#	7053#
	7057#	7078#	7094#	7095#	7100#	7101#	7106#	7126#	7142#	7143#	7148#	7149#	7153#	7173#	7189#
	7190#	7195#	7196#	7200#	7220#	7236#	7237#	7242#	7243#	7247#	7267#	7283#	7284#	7289#	7290#
	7294#	7315#	7331#	7332#	7337#	7338#	7342#	7362#	7378#	7379#	7384#	7385#	7389#	7411#	7427#
	7428#	7433#	7434#	7438#	7458#	7474#	7475#	7480#	7481#	7485#	7505#	7521#	7522#	7527#	7528#
	7532#	7552#	7568#	7569#	7574#	7575#	7579#	7599#	7615#	7616#	7621#	7622#	7626#	7646#	7662#
	7663#	7668#	7669#	7673#	7693#	7709#	7710#	7715#	7716#	7720#	7740#	7756#	7757#	7762#	7763#
	7767#	7787#	7803#	7804#	7809#	7810#	7814#	7835#	7851#	7852#	7857#	7858#	7862#	7882#	7898#
	7899#	7904#	7905#	7909#	7929#	7945#	7946#	7951#	7952#	7956#	7976#	7992#	7993#	7998#	7999#
	8003#	8023#	8039#	8040#	8045#	8046#	8050#	8070#	8086#	8087#	8092#	8093#	8097#		
M$TSTL	1921#	3706#	4282#	4455#	4456#	4457#	4458#	4459#	4460#	4461#	4462#	4463#	4464#	4465#	4467#
	4469#	4470#	4471#	4472#	4473#	4474#	4475#	4476#	4477#	4478#	4479#	4480#	4481#	4482#	4483#
	4484#	4485#	4486#	4487#	4488#	4491#	4492#	4493#	4523#	4551#	4554#	4557#	4561#	4574#	4652#
	4669#	4673#	4684#	4686#	4702#	4703#	4719#	4744#	4751#	4755#	4756#	4768#	4770#	4795#	4801#
	4802#	4807#	4810#	4815#	4816#	4821#	4822#	4834#	4841#	4842#	4847#	4848#	4851#	4852#	4869#
	4876#	4877#	4878#	4879#	4884#	4885#	4886#	4888#	4898#	4905#	4906#	4907#	4908#	4913#	4914#
	4916#	4926#	4933#	4934#	4935#	4936#	4941#	4942#	4944#	4955#	4963#	4964#	4969#	4970#	4973#
	4974#	4993#	4999#	5000#	5004#	5006#	5013#	5014#	5019#	5020#	5033#	5039#	5040#	5044#	5046#
	5054#	5055#	5060#	5061#	5073#	5079#	5080#	5087#	5088#	5093#	5094#	5099#	5100#	5112#	5119#
	5120#	5121#	5123#	5128#	5129#	5130#	5131#	5156#	5158#	5160#	5174#	5184#	5185#	5189#	5192#
	5203#	5204#	5209#	5210#	5223#	5233#	5234#	5238#	5241#	5252#	5253#	5258#	5259#	5272#	5282#
	5283#	5287#	5290#	5301#	5302#	5307#	5308#	5321#	5331#	5332#	5336#	5339#	5350#	5351#	5356#
	5357#	5370#	5384#	5385#	5390#	5393#	5408#	5409#	5415#	5416#	5429#	5443#	5445#	5450#	5453#
	5469#	5470#	5476#	5477#	5490#	5500#	5501#	5505#	5508#	5519#	5520#	5525#	5526#	5539#	5549#
	5550#	5554#	5557#	5568#	5569#	5574#	5575#	5588#	5598#	5599#	5603#	5606#	5617#	5618#	5623#
	5624#	5637#	5647#	5648#	5652#	5655#	5666#	5667#	5672#	5673#	5686#	5696#	5697#	5701#	5704#
	5715#	5716#	5721#	5722#	5735#	5745#	5746#	5750#	5753#	5764#	5765#	5770#	5771#	5784#	5794#
	5795#	5799#	5802#	5813#	5814#	5819#	5820#	5833#	5843#	5844#	5848#	5851#	5862#	5863#	5868#

```
                5869#  5882#  5900#  5901#  5906#  5909#  5918#  5919#  5925#  5926#  5939#  5949#  5950#  5954#  5957#
                5969#  5970#  5977#  5978#  5991#  5992#  6008#  6009#  6013#  6015#  6033#  6034#  6039#  6044#  6045#
                6058#  6077#  6078#  6083#  6086#  6098#  6099#  6104#  6105#  6114#  6120#  6124#  6126#  6128#  6131#
                6145#  6149#  6151#  6153#  6156#  6168#  6183#  6189#  6190#  6193#  6205#  6221#  6223#  6227#  6231#
                6239#  6261#  6262#  6263#  6266#  6295#  6296#  6297#  6300#  6328#  6329#  6330#  6333#  6381#  6383#
                6422#  6427#  6459#  6469#  6472#  6502#  6504#  6538#  6539#  6553#  6557#  6575#  6589#  6590#  6591#
                6592#  6614#  6630#  6631#  6636#  6637#  6641#  6660#  6676#  6677#  6682#  6683#  6687#  6706#  6722#
                6723#  6728#  6729#  6733#  6752#  6768#  6769#  6774#  6775#  6779#  6798#  6814#  6815#  6820#  6821#
                6825#  6844#  6860#  6861#  6866#  6867#  6871#  6890#  6906#  6907#  6912#  6913#  6917#  6936#  6952#
                6953#  6958#  6959#  6963#  6982#  6998#  6999#  7004#  7005#  7010#  7030#  7046#  7047#  7052#  7053#
                7057#  7078#  7094#  7095#  7100#  7101#  7106#  7126#  7142#  7143#  7148#  7149#  7153#  7173#  7189#
                7190#  7195#  7196#  7200#  7220#  7236#  7237#  7242#  7243#  7247#  7267#  7283#  7284#  7289#  7290#
                7294#  7315#  7331#  7332#  7337#  7338#  7342#  7362#  7378#  7379#  7384#  7385#  7389#  7411#  7427#
                7428#  7433#  7434#  7438#  7458#  7474#  7475#  7480#  7481#  7485#  7505#  7521#  7522#  7527#  7528#
                7532#  7552#  7568#  7569#  7574#  7575#  7579#  7599#  7615#  7616#  7621#  7622#  7626#  7646#  7662#
                7663#  7668#  7669#  7673#  7693#  7709#  7710#  7715#  7716#  7720#  7740#  7756#  7757#  7762#  7763#
                7767#  7787#  7803#  7804#  7809#  7810#  7814#  7835#  7851#  7852#  7857#  7858#  7862#  7882#  7898#
                7899#  7904#  7905#  7909#  7929#  7945#  7946#  7951#  7952#  7956#  7976#  7992#  7993#  7998#  7999#
                8003#  8023#  8039#  8040#  8045#  8046#  8050#  8070#  8086#  8087#  8092#  8093#  8097#
MSWORD          1994#  3706#  3747#  3770#  4282#  4514#  4751#  4768#  4801#  4815#  4841#  4847#  4876#  4884#  4905#
                4913#  4933#  4941#  4963#  4969#  4999#  5013#  5039#  5054#  5079#  5093#  5119#  5128#  5156#  5184#
                5203#  5233#  5252#  5282#  5301#  5331#  5350#  5384#  5408#  5443#  5469#  5500#  5519#  5549#  5568#
                5598#  5617#  5647#  5666#  5696#  5715#  5745#  5764#  5794#  5813#  5843#  5862#  5900#  5918#  5949#
                5969#  6008#  6033#  6077#  6098#  6126#  6128#  6151#  6153#  6190#  6223#  6227#  6261#  6263#  6295#
                6297#  6328#  6330#  6381#  6422#  6459#  6469#  6502#  6538#  6553#  6589#  6630#  6637#  6676#  6683#
                6722#  6729#  6768#  6775#  6814#  6821#  6860#  6867#  6906#  6913#  6952#  6959#  6998#  7005#  7046#
                7053#  7094#  7101#  7142#  7149#  7189#  7196#  7236#  7243#  7283#  7290#  7331#  7338#  7378#  7385#
                7427#  7434#  7474#  7481#  7521#  7528#  7568#  7575#  7615#  7622#  7662#  7669#  7709#  7716#  7756#
                7763#  7803#  7810#  7851#  7858#  7898#  7905#  7945#  7952#  7992#  7999#  8039#  8046#  8086#  8093#
                8123#  8124#  8125#  8126#  8183
MSXFER          1682#  3706#
OPEN            1171#  3706#
POINTE          1176#  3706#  3737
PRINTB          1239#  3706#  4455  4456  4457  4458  4459  4460  4461  4462  4463  4464  4465  4467  4469
                4470   4471   4472  4473  4474  4475  4476  4477  4478  4479  4480  4481  4482  4483  4484
                4485   4486   4487  4488
PRINTF          1279#  3706#  4491  4492
PRINTS          1319#  3706#
PRINTX          1359#  3706#
READBU          1399#  3706#
READEF          1403#  3706#  4551  4554  4557  4561
RERROR          4054#  6008   6033  6077  6098
RFLAGS          1408#  3706#
ROMCLK          4093#  4129   4131  4133  4139  4146  4153  4160  4168  4182  4243  4261  4263  4266  4271
                4275   4302   4313  4315  4323  4325  5177  5179  5196  5198  5226  5228  5245  5247  5275
                5277   5294   5296  5324  5326  5343  5345  5374  5376  5398  5400  5433  5435  5458  5460
                5493   5495   5512  5514  5542  5544  5561  5563  5591  5593  5610  5612  5640  5642  5659
                5661   5689   5691  5708  5710  5738  5740  5757  5759  5787  5789  5806  5808  5836  5838
                5855    857   5887  5895  5913  5942  5944  5961  5963  5996  6000  6002  6021  6025  6027
                6063   6068   6071  6090  6092  6122  6147  6181  6219  6251  6254  6284  6288  6290  6318
                6321   6352   6374  6376  6413  6415  6446  6449  6452  6463  6489  6492  6495  6531  6577
                6579   6581   6583  6618  6622  6624  6664  6668  6670  6710  6714  6716  6756  6760  6762
                6802   6806   6808  6848  6852  6854  6894  6898  6900  6940  6944  6946  6986  6990  6992
                7034   7038   7040  7082  7086  7088  7130  7134  7136  7177  7181  7183  7224  7228  7230
                7271   7275   7277  7319  7323  7325  7366  7370  7372  7415  7419  7421  7462  7466  7468
                7509   7513   7515  7556  7560  7562  7603  7607  7609  7650  7654  7656  7697  7701  7703
```

```
        7744    7748    7750    7791    7795    7797    7859    7843    7845    7886    7890    7892    7933    7937    7939
        7980    7984    7986    8027    8031    8033    8074    8078    8080
SETPRI  1413#   3706#   6120    6124    6145    6149    6168    6183    6205    6221
SETVEC  1418#   3706#
SLASH   1424#   3706#
STARS   1438#   3706#
SVC     1452#   3705#   3706
XFER    1612#   3706#   4514#   6263#   6297#   6330#   6553#   6637#   6683#   6729#   6775#   6821#   6867#   6913#   6959#
        7005#   7053#   7101#   7149#   7196#   7243#   7290#   7338#   7385#   7434#   7481#   7528#   7575#   7622#   7669#
        7716#   7763#   7810#   7858#   7905#   7952#   7999#   8046#   8093#
XFERF   1616#   3706#
XFERT   1620#   3706#
$MD     4439#   4455    4456    4457    4458    4459    4460    4461    4462    4463    4464    4465    4467    4469    4470
        4471    4472    4473    4474    4475    4476    4477    4478    4479    4480    4481    4482    4483    4484    4485
        4486    4487    4488


. ABS.  040004      000


 ERRORS DETECTED:  0

 CZDMPC.BIC,CZDMPC.SEQ/CRF/DOC/NL:TOC=SYC34R.MLB,CZDMPC.P11
 RUN-TIME: 46 55 6 SECONDS
 RUN-TIME RATIO: 177/108=1.6
 CORE USED:  21K  (41 PAGES)

 DOCUMENT PAGES:  236
```