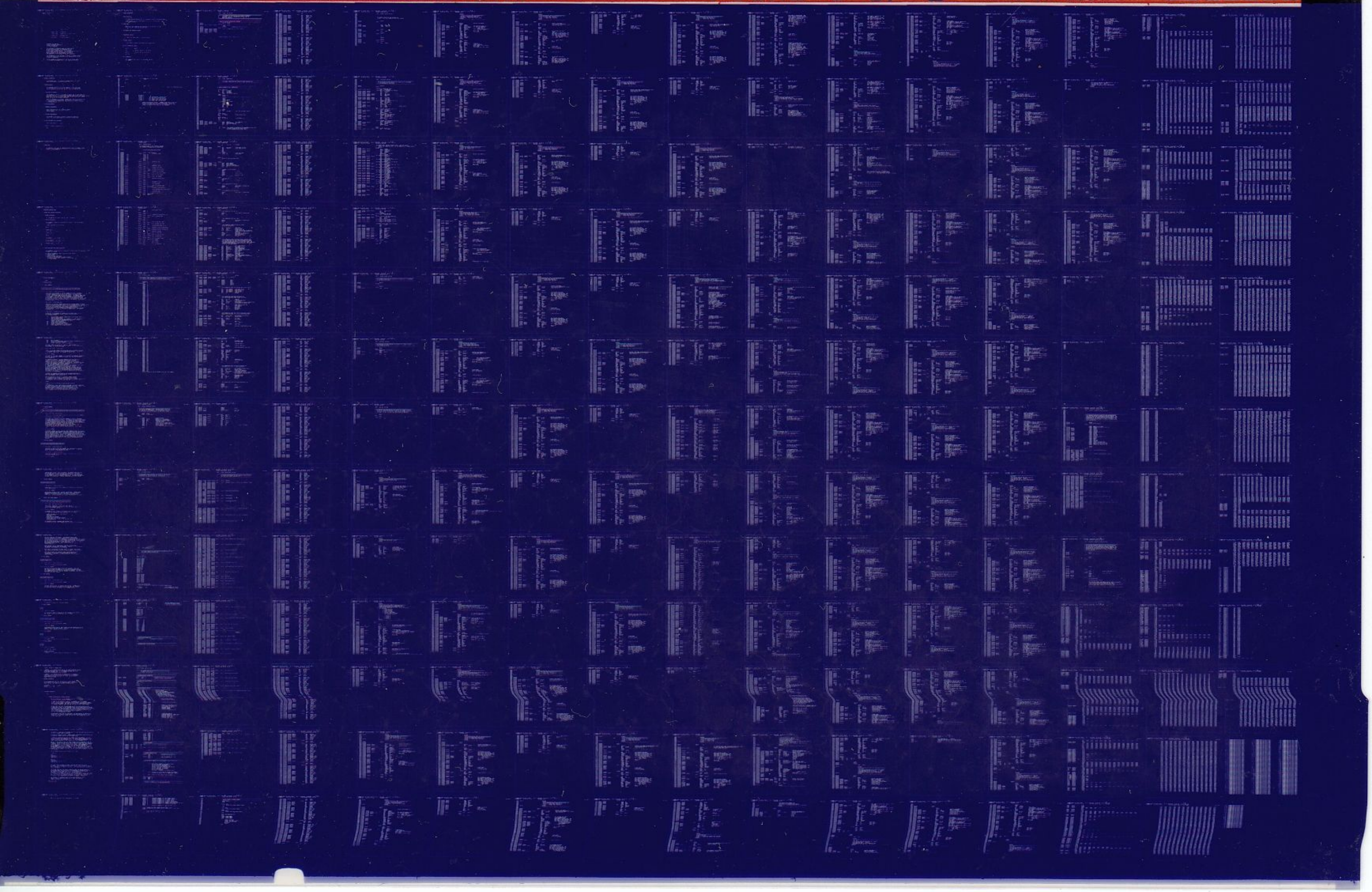
KMC11-B

KMC11-B STATIC PART 1 AH-S877A-MC
CZKMBAO FICHE 1 OF 2



KMC11-B

KMC11-B STATIC PART 1 AH-S877A-MC
CZKMBA0 FICHE 2 OF 2

```
        .NLIST  SEQ,BIN,LOC
        .REM &
```

IDENTIFICATION

PRODUCT CODE: AC-S875A-MC

PRODUCT NAME: CZKMBA0 KMC11-B STATIC PART1

PROGRAM DATE: SEPTEMBER 1981

MAINTAINER: CSS/NSG DIAGNOSTICS

1.0 GENERAL INFORMATION

THIS PROGRAM TESTS 1 TO 64 KMC11-B (M8206) MODULES. IT RUNS
UNDER THE DIAGNOSTIC RUNTIME SERVICES (DRS).

1.1 PROGRAM ABSTRACT

THIS PROGRAM CONSISTS OF A SET OF SEQUENTIAL LOGIC TESTS USED
TO VERIFY MOST OF THE LOGIC OF THE KMC11-B. IT IS RUN BEFORE,
AND IN CONJUNCTION WITH, CZKMC TO FULLY CHECK THE KMC11-B LOGIC.

1.1.1 STRUCTURE OF PROGRAM

THIS DIAGNOSTIC OCCUPIES 14.5K WORDS OF MEMORY AND IS COMPATIBLE WITH
BOTH XXDP+ AND ACT. IT CAN BE RUN STANDALONE UNDER XXDP+, AND CAN BE CHAINED
UNDER XXDP+, ACT AND APT IN ACT MODE (SEE "CREATE CORE IMAGE" COMMAND
BELOW FOR DETAILS OF CHAINING PROCEDURE).

WHEN THIS DIAGNOSTIC IS STARTED, CONTROL GOES FIRST TO THE SUPERVISOR.
IT WILL THEN ENTER COMMAND MODE, INDICATED BY THE PROMPT 'DR>' .
AT COMMAND MODE THE OPERATOR MAY THEN ENTER ANY OF SEVERAL COMMANDS
AS DESCRIBED BELOW.

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

PDP-11 PROCESSOR WITH 16K OR MORE OF MEMORY
CONSOLE DEVICE (LA36,LA120,LA34,VT100,ETC.)
KMC11-B (M8206)

1.2.2 SOFTWARE REQUIREMENTS

THE PROGRAM IS REVISION-D DIAGNOSTIC SUPERVISOR COMPATIBLE.
CONSULT THE XXDP+ USERS MANUAL FOR OPERATING INSTRUCTIONS.

1.3 RELATED DOCUMENTS AND STANDARDS

XXDP+ USERS MANUAL CHQUS

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

NONE

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE SUBSYSTEM BEING TESTED IS ASSUMED TO WORK PROPERLY. FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, MEMORY, ETC., DO NOT FUNCTION PROPERLY.

2.0 OPERATING INSTRUCTIONS

2.1 LOADING AND STARTING PROCEDURES

2.1.1 LOADING PROCEDURES

THIS PROGRAM MAY BE LOADED FROM ANY XXDP+ LOAD MEDIA.

2.1.2 STARTING PROCEDURES

THE DIAGNOSTIC SUPERVISOR AUTOSTARTS ON LOADING. IT MAY BE RESTARTED AT 200(8).

A SAMPLE DIALOGUE IS LISTED BELOW:

```
DR>STA

CHANGE HW (L)  ? Y

# UNITS (D)  ? 1

UNIT 0

CSR ADDRESS :  (O)  174100 ? <CR>

VECTOR ADDRESS :  (O)  300 ? <CR>

PRIORITY LEVEL :  (O)  5 ? <CR>
```

2.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

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

A) LOAD THE DIAGNOSTIC FROM THE RELEVANT XXDP+ MEDIUM.
B) RECEIVE PROMPT 'DR>'
C) ENTER STA<CR>
D) ANSWER HARDWARE QUESTIONS
E) GET END OF PASS MESSAGES OR ERROR MESSAGES
F) TO END EXECUTION, ENTER CONTROL/C. THIS RETURNS THE 'DR>' PROMPT.

2.2 SPECIAL ENVIRONMENTS

NONE

2.3 PROGRAM OPTIONS

2.3.1 START COMMAND

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

2.3.1.1 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 2.3.1.

2.3.1.2 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 2.3.1.

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

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

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

BOE BELL ON ERROR
UAM RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
ISR INHIBIT STATISTICAL REPORTS
IDR INHIBIT DROPPING OF UNITS BY DIAGNOSTIC

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.

2.3.1.4 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 2.3.1.

2.3.1.5 EFFECT OF 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 64. THE TERM "UNIT" REFERS TO THE DEVICE TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED. FOLLOWING THIS ARE THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES WILL BE BUILT. EACH P-TABLE IS A CORE-RESIDENT TABLE CONTAINING ALL THE HARDWARE INFORMATION FOR ONE UNIT. THE OPERATOR MUST SUPPLY N (NUMBER OF UNITS) VALUES FOR EACH QUESTION. HE MAY DO THIS BY GIVING ONE ANSWER TO EACH QUESTION (IN WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR BY GIVING N VALUES, SEPARATED BY COMMAS, TO EACH QUESTION (SERIES WILL BE POSED ONCE). EACH QUESTION IS FOLLOWED BY THE RESPONSE RADIX (D FOR DECIMAL, B FOR BINARY, O FOR OCTAL, L FOR YES/NO) IN PARENTHESES AND THE DEFAULT VALUE AFTER THE PARENTHESES.

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

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

EXAMPLE:

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

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

2.3.2 RESTART COMMAND

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

2.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

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

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

<UNIT-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1,2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5, 8-10 ETC.) THAT SPECIFY THE UNITS TO BE TESTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS MAY RANGE FROM 1 THRU N (N IS THE NUMBER OF UNITS SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS ENTERED DURING THE HARDWARE DIAGLOGUE. THE UNITS WHICH ARE SELECTED MUST NOT HAVE BEEN DROPPED BY THE DROP COMMAND. SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP COMMAND.

2.3.2.3 EFFECT OF 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 RE-EXECUTED (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.

2.3.3 CONTINUE COMMAND

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

2.3.3.1 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.

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

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

2.3.3.3 EFFECT OF 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.

2.3.4 PROCEED COMMAND

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

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

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

2.3.4.2 EFFECT OF 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.

2.3.5 CREATE CORE IMAGE COMMAND

****
CCI/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:<FLAG-LIST>
****

2.3.5.1 TESTS, PASS, AND FLAGS SWITCHES

<TEST-LIST>, <PASS-CNT>, <FLAG-LIST>, AND ARE AS IN THE START COMMAND, EXCEPT THAT THE UAM (UNATTENDED MODE) FLAG DEFAULTS TO THE SET POSITION.

2.3.5.2 EFFECT OF COMMAND

THE PURPOSE OF THIS COMMAND IS TO CREATE A BIC FILE SUITABLE FOR CHAIN MODE EXECUTION. THE XXDP+ PROCEDURE IS AS FOLLOWS:

```
INVOKE THE XXDP+ UTILITY UPD1 OR UPD2
LOAD XXN:FILE.BIN
START 200
<QUESTIONS AND ANSWERS>
RESTART UPD1 USING RESTART ADDRESS
HICORE ADDRESS (IF "PASSED 14.5K" MESSAGE CAME)
DUMP XXN:FILE.BIC
```

THE OPERATOR DIALOGUE (HARDWARE AND SOFTWARE) WILL BE EXECUTED AS IN THE START COMMAND, BUT AT THE END OF THE QUESTIONS

THE HALT STATE WILL BE ENTERED. THE OPERATOR SHOULD THEN DUMP THE PROGRAM TO THE XXDP+ LIBRARY USING A BIC EXTENSION TO INDICATE THAT THIS FILE IS CHAINABLE. HE SHOULD USE THE XXDP+ UTILITY UPD1 OR UPD2 TO DO THIS. IF THE P-TABLES EXTEND BEYOND 14.5K, A MESSAGE WILL BE ISSUED GIVING THE NEW UPPER CORE LIMIT, TO WHICH THE OPERATOR MUST ADJUST BEFORE DUMPING. HE MAY NOW DELETE THE NON-CHAINABLE BIN FILE IF DESIRED, SINCE THE BIC FILE HAS ALL THE CAPABILITIES OF IT.

WHEN THIS BIC FILE IS SUBSEQUENTLY EXECUTED IN CHAIN MODE, THE OPERATOR DIALOGUES WILL BE BYPASSED. HOWEVER, IF IT IS EXECUTED STANDALONE, THE DIALOGUE WILL BE REISSUED.

NOTE THAT IF THE MESSAGE "TOO MANY UNITS" IS ISSUED, TWO OR MORE CORE IMAGES MUST BE CREATED (WITH DIFFERENT NAMES) TO TEST ALL UNITS.

NOTE THAT ALTHOUGH THE CHAINABLE IMAGE CAN BE EXECUTED ON A 16K MACHINE, THE ORIGINAL CCI CREATION MUST BE DONE ON A LARGE MACHINE, THE EXACT SIZE BEING DEPENDENT ON WHICH UPDATE UTILITY IS USED.

2.3.6 ADD COMMAND

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

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

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

2.3.6.2 EFFECT OF 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.

2.3.7 DROP COMMAND

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

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

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

2.3.7.2 EFFECT OF 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.

2.3.8 PRINT COMMAND

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

2.3.8.1 EFFECT OF COMMAND

ALL STATISTICS TABLES ACCUMULATED BY THE DIAGNOSTIC ARE PRINTED. THE ISR (INHIBIT STATISTICAL REPORTING) FLAG IS CLEARED.

2.3.9 DISPLAY COMMAND

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

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

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

2.3.9.2 EFFECT OF 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.

2.3.10 FLAGS COMMAND

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

2.3.10.1 EFFECT OF COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

2.3.11 ZFLAGS COMMAND

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

2.3.11.1 EFFECT OF COMMAND

ALL FLAGS ARE CLEARED.

2.3.12 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 (HARD CORE QUESTIONS (SEE 1.1.1),HARDWARE DIALOGUE (SEE 2.3.1.5), OR SOFTWARE DIALOGUE (SEE 2.3.1.5) 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.

2.3.13 HARDWARE PARAMETERS

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

```
CSR ADDRESS     (O)     174100 ?
VECTOR          (O)     300    ?
PRIORITY LEVEL  (O)     5      ?
```

2.3.14 SOFTWARE PARAMETERS

NONE

2.3.15 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 64 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 64 TABLES. LET THE DESIRED VALUE FOR THE SECOND
PARAMETER BE EQUAL TO THE UNIT NUMBER (1,2,3,...,64) EXCEPT FOR UNIT
50, WHICH SHOULD RECEIVE THE VALUE 49. LET THE DESIRED VALUE FOR THE THIRD
PARAMETER BE THE NUMBER 76 FOR THE FIRST 20 UNITS AND THE NUMBER 77 FOR THE
LAST 44 UNITS.

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

```
# UNITS (D)  ?  64

UNIT 1
<QUESTION 1> ? 75
<QUESTION 2> ? 1-20
<QUESTION 3> ? 76

UNIT 21
<QUESTION 1> ?
<QUESTION 2> ? 21-49,,51-64
<QUESTION 3> ? 77
```

THE FIRST TIME THE SERIES IS ASKED, SLOT ONE RECEIVES A 75 IN ALL
64 TABLES. SLOT TWO RECEIVES THE VALUES 1,2,3,...,20 IN TABLES 1 THRU 20
AND A CONSTANT 20 IN TABLES 21 THRU 64. SLOT THREE RECEIVES A CONSTANT 76
IN ALL 64 TABLES.

THE SECOND TIME THRU THE SERIES, TABLES 21 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 21 THRU 64, SINCE NO NEW EXLICIT VALUES ARE TYPED
IN. SLOT TWO GETS THE VALUES 21,22,23,...,49 IN TABLES 21 THRU 49, AND
GETS A 49 IN SLOT 50, AND GETS THE VALUES 51,52,53,...,64 IN TABLES
51 THRU 64. SLOT THREE GETS THE VALUE 77 IN TABLES 21 THRU 64.

THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT 64
EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ONE QUESTION
(NAMELY QUESTION 2).

2.4 EXECUTION TIMES

ONE PASS OF ONE UNIT TAKES APPROXIMATELY 10 SECONDS.

3.0 ERROR INFORMATION

3.1 ERROR REPORTING

THE ERROR MESSAGES PRODUCED BY THIS DIAGNOSTIC HAVE THE FOLLOWING FORMAT:

CZKMBA DEV[FTL OR HRD] ERROR NNNNN ON UNIT UU TST NN SUB NNN PC XXXXX
ASCII ERROR MESSAGE

3.2 ERROR HALTS

ERROR HALTS ARE SUPPORTED PER DESCRIBED IN THE PREVIOUS SECTION
WITH /FLAG:HOE. THERE ARE NO OTHER HALTS.

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

A CUMALATIVE ERROR COUNT IS GIVEN AFTER PASS.

4.2 PROGRESS REPORTS

A PASS COUNT IS UPDATED AND PRINTED AFTER EVERY PASS.

5.0 DEVICE INFORMATION TABLES

DEVICE:	M8206
CSR:	FLOATING (174100 IS THE DEFAULT)
VECTOR:	FLOATING (300 IS THE DEFAULT)
BR LEVEL:	5

6.0 TEST SUMMARIES

TEST SUMMARIES ARE GIVEN AT THE BEGINNING OF EACH
TEST MODULE

SEQ 0015

```
                        .TITLE CZKMBA0 KMC11-B STATIC PART1
002000                  .=2000



                        .MCALL  SVC
002000                  SVC                             ; INITIALIZE SUPERVISOR MACROS


002000                  BGNMOD  CZKMB

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.
                        ;--

002000                          POINTER BGNAU,BGNDU,BGNSW,BGNRPT


                        HEADER  CZKMB,A,0,240.,0
002000                  L$NAME::                ;DIAGNOSTIC NAME
002000  103                     .ASCII /C/
002001  132                     .ASCII /Z/
002002  113                     .ASCII /K/
002003  115                     .ASCII /M/
002004  102                     .ASCII /B/
002005  000                     .BYTE   0
002006  000                     .BYTE   0
002007  000                     .BYTE   0
002010                  L$REV::                 ;REVISION LEVEL
002010  101                     .ASCII  /A/
002011                  L$DEPO::                ;0
002011  060                     .ASCII  /0/
002012                  L$UNIT::                ;NUMBER OF UNITS
002012  000000                  .WORD   0
002014                  L$TIML::                ;LONGEST TEST TIME
002014  000360                  .WORD   240.
002016                  L$HPCP::                ;POINTER TO H.W. QUES.
002016  041146                  .WORD   L$HARD
002020                  L$SPCP::                ;POINTER TO S.W. QUES.
002020  000000                  .WORD   0
002022                  L$HPTP::                ;PTR. TO DEF. H.W. PTABLE
002022  002366                  .WORD   L$HW
002024                  L$SPTP::                ;PTR. TO S.W. PTABLE
002024  002412                  .WORD   L$SW
002026                  L$LADP::                ;DIAG. END ADDRESS
002026  041740                  .WORD   L$LAST
002030                  L$STA::                 ;RESERVED FOR APT STATS
002030  000000                  .WORD   0
002032                  L$CO::
002032  000000                  .WORD   0
002034                  L$DTYP::                ;DIAGNOSTIC TYPE
002034  000000                  .WORD   0
002036                  L$APT::                 ;APT EXPANSION
002036  000000                  .WORD   0
002040                  L$DTP::                 ;PTR. TO DISPATCH TABLE
002040  002124                  .WORD   L$DISPATCH
002042                  L$PRIO::                ;DIAGNOSTIC RUN PRIORITY
002042  000000                  .WORD   0
002044                  L$ENVI::                ;FLAGS DESCRIBE HOW IT WAS SETUP
002044  000000                  .WORD   0
002046                  L$EXP1::                ;EXPANSION WORD
002046  000000                  .WORD   0
002050                  L$MREV::                ;SVC REV AND EDIT #
002050  003                     .BYTE   C$REVISION
002051  003                     .BYTE   C$EDIT
```

```
002052                  L$EF::                          ;DIAG. EVENT FLAGS
002052  000000                  .WORD   0
002054  000000                  .WORD   0
002056                  L$SPC::
002056  000000                  .WORD   0
002060                  L$DEVP::                        ; POINTER TO DEVICE TYPE LIST
002060  003116                  .WORD   L$DVTYP
002062                  L$REPP::                        ;PTR. TO REPORT CODE
002062  012454                  .WORD   L$RPT
002064                  L$EXP4::
002064  000000                  .WORD   0
002066                  L$EXP5::
002066  000000                  .WORD   0
002070                  L$AUT::                         ;PTR. TO ADD UNIT CODE
002070  013174                  .WORD   L$AU
002072                  L$DUT::                         ;PTR. TO DROP UNIT CODE
002072  013170                  .WORD   L$DU
002074                  L$LUN::                         ;LUN FOR EXERCISERS TO FILL
002074  000000                  .WORD   0
002076                  L$DESP::                        ;POINTER TO DIAG. DESCRIPTION
002076  003124                  .WORD   L$DESC
002100                  L$LOAD::                        ;GENERATE SPECIAL AUTOLOAD EMT
002100  104035                  EMT     E$LOAD
002102                  L$ETP::                         ;POINTER TO ERRTBL
002102  000000                  .WORD   0
002104                  L$ICP::                         ;PTR. TO INIT CODE
002104  012472                  .WORD   L$INIT
002106                  L$CCP::                         ;PTR. TO CLEAN-UP CODE
002106  013166                  .WORD   L$CLEAN
002110                  L$ACP::                         ;PTR. TO AUTO CODE
002110  012470                  .WORD   L$AUTO
002112                  L$PRT::                         ;PTR. TO PROTECT TABLE
002112  012462                  .WORD   L$PROT
002114                  L$TEST::                        ;TEST NUMBER
002114  000000                  .WORD   0
002116                  L$DLY::                         ;DELAY COUNT
002116  000000                  .WORD   0
002120                  L$HIME::                        ;PTR. TO HIGH MEM
002120  000000                  .WORD   0
```

```
                                .SBTTL  DISPATCH TABLE

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

002122                                  DISPATCH        80.
002122  000120                          .WORD   80
002124                          L$DISPATCH::
002124  013176                          .WORD   T1
002126  013320                          .WORD   T2
002130  013366                          .WORD   T3
002132  013576                          .WORD   T4
002134  013746                          .WORD   T5
002136  014112                          .WORD   T6
002140  014252                          .WORD   T7
002142  014412                          .WORD   T8
002144  014564                          .WORD   T9
002146  014770                          .WORD   T10
002150  015174                          .WORD   T11
002152  015376                          .WORD   T12
002154  015562                          .WORD   T13
002156  015674                          .WORD   T14
002160  016134                          .WORD   T15
002162  016430                          .WORD   T16
002164  016670                          .WORD   T17
002166  017130                          .WORD   T18
002170  017370                          .WORD   T19
002172  017630                          .WORD   T20
002174  020120                          .WORD   T21
002176  020422                          .WORD   T22
002200  020662                          .WORD   T23
002202  021122                          .WORD   T24
002204  021362                          .WORD   T25
002206  021622                          .WORD   T26
002210  022062                          .WORD   T27
002212  022322                          .WORD   T28
002214  022562                          .WORD   T29
002216  023022                          .WORD   T30
002220  023330                          .WORD   T31
002222  023726                          .WORD   T32
002224  024412                          .WORD   T33
002226  024722                          .WORD   T34
002230  025262                          .WORD   T35
002232  025570                          .WORD   T36
002234  025736                          .WORD   T37
002236  026100                          .WORD   T38
002240  026252                          .WORD   T39
002242  026456                          .WORD   T40
002244  026632                          .WORD   T41
002246  027006                          .WORD   T42
002250  027156                          .WORD   T43
002252  027350                          .WORD   T44
002254  027570                          .WORD   T45
002256  030016                          .WORD   T46
```

```
872  002260  030154                    .WORD   T47
873  002262  030376                    .WORD   T48
874  002264  030542                    .WORD   T49
875  002266  030752                    .WORD   T50
876  002270  031162                    .WORD   T51
877  002272  031372                    .WORD   T52
878  002274  031602                    .WORD   T53
879  002276  032012                    .WORD   T54
880  002300  032222                    .WORD   T55
881  002302  032432                    .WORD   T56
882  002304  032642                    .WORD   T57
883  002306  033054                    .WORD   T58
884  002310  033264                    .WORD   T59
885  002312  033474                    .WORD   T60
886  002314  033704                    .WORD   T61
887  002316  034114                    .WORD   T62
888  002320  034324                    .WORD   T63
889  002322  034534                    .WORD   T64
890  002324  034744                    .WORD   T65
891  002326  035154                    .WORD   T66
892  002330  035364                    .WORD   T67
893  002332  035574                    .WORD   T68
894  002334  036004                    .WORD   T69
895  002336  036214                    .WORD   T70
896  002340  036424                    .WORD   T71
897  002342  036634                    .WORD   T72
898  002344  037044                    .WORD   T73
899  002346  037254                    .WORD   T74
900  002350  037464                    .WORD   T75
901  002352  037674                    .WORD   T76
902  002354  040104                    .WORD   T77
903  002356  040314                    .WORD   T78
904  002360  040524                    .WORD   T79
905  002362  040734                    .WORD   T80
906                                    ;LNT.ED DIFINED AT END OF PROGRAM TO BE LAST TEST NUMBER.
907
908
909
910
911
912
```

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

002364                          BGNHW   DFPTBL
002364  000011                  .WORD   L10000-L$HW/2
002366                  L$HW::
002366                  DFPTBL::

002366  000006                  .WORD   6               ;MICRO CPU TYPE
002370  174100                  .WORD   174100          ;M8200,4,6,7 CSR ADDRESS
002372  000300                  .WORD   300             ;M8200,4,6,7 VECTOR ADDRESS
002374  005000                  .WORD   5000            ;INTERRUPT PRIORITY LEVEL
002376  000003                  .WORD   3               ;LINE UNIT TYPE
002400  000056                  .WORD   56              ;SWITCH PACK #1 (DDCMP LINE #)
002402  000000                  .WORD   0               ;SWITCH PACK #2 (BM873 BOOT ADDRESS)
002404  000000                  .WORD   0               ;SWITCH PACK #3
002406  000000                  .WORD   0               ;TEST CONNECTOR INSTALLED FLAG

002410                          ENDHW
002410                  L10000:
```

```
                        .SBTTL  SOFTWARE P-TABLE

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

002410                          BGNSW   SFPTBL
002410  000000                  .WORD   L10001-L$SW/2
002412                  L$SW::
002412                  SFPTBL::


002412                          ENDSW
002412                  L10001:
```

```
                        .SBTTL  GLOBAL EQUATES SECTION

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

002412                          EQUALS
                        ;
                        ; BIT DIFINITIONS
                        ;
        100000          BIT15== 100000
        040000          BIT14== 40000
        020000          BIT13== 20000
        010000          BIT12== 10000
        004000          BIT11== 4000
        002000          BIT10== 2000
        001000          BIT09== 1000
        000400          BIT08== 400
        000200          BIT07== 200
        000100          BIT06== 100
        000040          BIT05== 40
        000020          BIT04== 20
        000010          BIT03== 10
        000004          BIT02== 4
        000002          BIT01== 2
        000001          BIT00== 1
                        ;
        001000          BIT9==  BIT09
        000400          BIT8==  BIT08
        000200          BIT7==  BIT07
        000100          BIT6==  BIT06
        000040          BIT5==  BIT05
        000020          BIT4==  BIT04
        000010          BIT3==  BIT03
        000004          BIT2==  BIT02
        000002          BIT1==  BIT01
        000001          BIT0==  BIT00
                        ;
                        ; EVENT FLAG DEFINITIONS
                        ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                        ;
        000040          EF.START==      32.                             ; START COMMAND WAS ISSUED
        000037          EF.RESTART==    31.                             ; RESTART COMMAND WAS ISSUED
```

```
1020            000036          EF.CONTINUE==   30.             ; CONTINUE COMMAND WAS ISSUED
1021            000035          EF.NEW==        29.             ; A NEW PASS HAS BEEN STARTED
1022            000034          EF.PWR==        28.             ; A POWER-FAIL/POWER-UP OCCURRED
1023                            ;
1024                            ;
1025                            ; PRIORITY LEVEL DEFINITIONS
1026                            ;
1027            000340          PRI07== 340
1028            000300          PRI06== 300
1029            000240          PRI05== 240
1030            000200          PRI04== 200
1031            000140          PRI03== 140
1032            000100          PRI02== 100
1033            000040          PRI01== 40
1034            000000          PRI00== 0
1035                            ;
1036                            ;OPERATOR FLAG BITS
1037                            ;
1038            000004          EVL==       4
1039            000010          LOT==      10
1040            000020          ADR==      20
1041            000040          IDU==      40
1042            000100          ISR==     100
1043            000200          UAM==     200
1044            000400          BOE==     400
1045            001000          PNT==    1000
1046            002000          PRI==    2000
1047            004000          IXE==    4000
1048            010000          IBE==   10000
1049            020000          IER==   20000
1050            040000          LOE==   40000
1051            100000          HOE==  100000
1052
1053
1054
1055
1056
1057                            ;*******************************************************************************
1058                            ;* INSTRUCTION DEFINITIONS
1059                            ;*******************************************************************************
1060            022626          POP2SP=22626            ;INCREMENT STACK TWICE
1061
1062
1063
1064                            ;;*******************************************************************************
1065                            ;* PROGRAM EVENT FLAG DEFINITIONS
1066                            ;;*******************************************************************************
1067
1068
1069
```

```
                                .SBTTL  GLOBAL DATA SECTION

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

                                ;*******************************************************************************
                                ;* PROGRAM CONTROL PARAMETERS
                                ;*******************************************************************************
002412  000000                  NEXT:   .WORD   0               ;ADDRESS OF NEXT TEST TO BE EXECUTED
002414  000000                  LOCK:   .WORD   0               ;ADDRESS FOR LOCK CURRENT DATA

                                ;*******************************************************************************
                                ;* BUFFERS FOR INPUT-OUTPUT
                                ;*******************************************************************************
002416  000000                  TEMP:   0
        002460                  .=.+40
002460  000000                  MDATA:  0
        002522                  .=.+40

                                ;*******************************************************************************
                                ;* MISCELLANEOUS STORAGE
                                ;*******************************************************************************
002522  000000                  $TMPO:  .WORD   0               ;SCRATCH STORAGE
002524  000000                  LOGDEV: .WORD   0               ;LOGICAL DEVICE NUMBER
002526  000000                  PSTACK: .WORD   0               ;BASE LEVEL PROGRAM STACK POINTER
002530  000000                  SUBRPC: .WORD   0               ;PC OF SUBR CALL FOR ERROR REPORTS
002532  000000                  ERRFLG: .WORD   0               ;SUBROUTINE ERROR FLAG
002534  000000                  RETADR: .WORD   0               ;SUBR ERROR RETURN ADDRESS
002536  000000                  STRTSW: .WORD   0               ;SWITCHES AT START OF PROGRAM
002540  000000                  STAT:   .WORD   0               ;M8200,4,6,7 STATUS WORD STORAGE
002542  000000                  CLKX:   .WORD   0               ;
002544  000000                  MASKX:  .WORD   0               ;
002546  000000                  SAVSP:  .WORD   0               ;STACK POINTER STORAGE
002550  000000                  SAVPC:  .WORD   0               ;PROGRAM COUNTER STORAGE
002552  000000                  ZERO:   .WORD   0               ;
002554  000001                  ONE:    .WORD   1               ;
002556  041732                  MEMLIM: .WORD   MEMEND          ;HIGHEST LOCATION FOR NPR'S
002560                          MEMSZ:
002560  000000                          .WORD   0
002562  000001                  KMACTV: .BLKW   1               ;M8200,4,6,7 SELECTED ACTIVE
002564  000001                  KMNUM:  .BLKW   1               ;OCTAL NUMBER OF M8200,4,6,7
002566  000001                  SAVACT: .BLKW   1               ;ORIGINAL ACTIVE DEVICES
002570  000001                  SAVNUM: .BLKW   1               ;WORKABLE NUMBER
002572  000000                  FLAG:   .WORD   0               ;SCRATCH STORAGE
002574  000000                  RUN:    .WORD   0               ;POINTER TO RUNNING DEVICES
002576  000000                  MRO:    .WORD   0
002600  000006                  WTYPE:  .WORD   6
002602  000000                  TYPE:   .WORD   0
002604  000000                  PONE:   .WORD   0               ; FIRST PASS SWITCH
002606  000000                  $GDADR: .WORD   0               ;CONTAINS ADDRESS OF 'GOOD' DATA
002610  000000                  $BDADR: .WORD   0               ;CONTAINS ADDRESS OF 'BAD' DATA
002612  000000                  $GDDAT: .WORD   0               ;CONTAINS 'GOOD' DATA
002614  000000                  $BDDAT: .WORD   0               ;CONTAINS 'BAD' DATA
002616  000000                          .WORD   0               ;RESERVED--NOT TO BE USED
```

```
1126  002620  000000                          .WORD   0
1127  002622  000000                  FTIME:  .WORD   0
1128  002624  000000                  SAVE4:  .WORD   0
1129  002626  000000                  SAVE6:  .WORD   0
1130  002630  000000                  SCLK:   .WORD   0
1131  002632  000000                  SIBS10: .WORD   0
1132  002634  000000                  SIBS11: .WORD   0
1133  002636  000000                  SIBS12: .WORD   0
1134  002640  000000                  SIBS13: .WORD   0
1135
1136                                  ;*****************************************************************************
1137                                  ;* DATA PATTERNS
1138                                  ;*****************************************************************************
1139  002642    000   377   000       MEMDAT: .BYTE   0,-1,0,-1,125,252,125,252
1140  002645    377   125   252
1141  002650    125   252
1142  002652    000   000   377       SPDAT:  .BYTE   0,0,-1,-1,125,125,252,252
1143  002655    377   125   125
1144  002660    252   252
1145                                          .EVEN
1146
1147                                  ;*****************************************************************************
1148                                  ;* PROGRAM CONTROL FLAGS
1149                                  ;*****************************************************************************
1150  002662    000                   INIFLG: .BYTE   0               ;PROGRAM INITIALIZING FLAG
1151        002664                            .EVEN
1152  002664    000                   LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
1153  002665    000                   QV.FLG: .BYTE   0               ;QUICK VERIFY FLAG
1154                                          .EVEN
1155
1156                                  ;*****************************************************************************
1157                                  ;* DEFINITION OF M8200,4,6,7 STATUS WORDS - STAT1,STAT2,STAT3
1158                                  ;*
1159                                  ;*
1160                                  ;*      STAT1 - BITS 00-08 IS M8200,4,6,7 VECTOR ADDRESS
1161                                  ;*              BIT15=1 LINE UNIT IS AN M8203
1162                                  ;*              BIT14=0 NO TEST CONNECTOR(S) USED
1163                                  ;*              BIT14=1 H-XXX TEST CONNECTOR WILL BE USED
1164                                  ;*              BIT13=0 LINE UNIT IS AN M8201
1165                                  ;*              BIT13=1 LINE UNIT IS AN M8202
1166                                  ;*              BIT12=1 NO LINE UNIT
1167                                  ;*              BITS 09-11 IS M8200,4,6,7 PRIORITY LEVEL
1168                                  ;*
1169                                  ;*      STAT2 - LOW BYTE IS SWITCH PACK #1 (DDCMP LINE NUMBER)
1170                                  ;*              HIGH BYTE IS SWITCH PACK #2 (BM873 BOOT ADDRESS)
1171                                  ;*
1172                                  ;*      STAT3 - BIT0=1 DO FREE RUNNING TESTS ON M8200,4,6,7
1173                                  ;*
1174                                  ;*****************************************************************************
1175  002666  000000                  STAT1:  .WORD   0
1176  002670  000000                  STAT2:  .WORD   0
1177  002672  000000                  STAT3:  .WORD   0
1178
1179                                  ;*****************************************************************************
1180                                  ;* POINTERS TO M8200,4,6,7 VECTORS AND REGISTERS
1181                                  ;*****************************************************************************
```

```
1182  002674  000000                KMRVEC: 0       ;POINTER TO M8200,4,6,7 RCV INTRPT VECTOR
1183  002676  000000                KMRLVL: 0       ;POINTER TO M8200,4,6,7 RCV INTRPT SERVICE PS
1184  002700  000000                KMTVEC: 0       ;POINTER TO M8200,4,6,7 TX INTRPT VECTOR
1185  002702  000000                KMTLVL: 0       ;POINTER TO M8200,4,6,7 TX INTRPT SERVICE PS
1186  002704  000000                KMCSR:  0       ;POINTER TO M8200,4,6,7 CONTROL STATUS REGISTER
1187  002706  000000                KMCSRH: 0       ;POINTER TO M8200,4,6,7 CONTROL STATUS REGISTER HIGH BYTE
1188  002710  000000                KMCTL:  0       ;POINTER TO M8200,4,6,7 CONTROL OUT REGISTER
1189  002712  000000                KMP04:  0
1190  002714  000000                KMP06:  0       ;POINTER TO M8200,4,6,7 PORT REGISTER - SEL6
1191
1192                                ;;****  PRIMARY REG ADRS STORAGE FOR THIS UNIT  ******
1193                                ;THESE LOCATIONS WILL BE LOADED FOR THE CURRENT UNIT, IN INIT CODE
1194  002716                        REGADR:
1195
1196                                ;;****  STACK USED FOR SUBROUTINE LINKAGE  *****
1197  002716  000100                        .BLKW   100
1198  003116                        SSTACK:
1199
1200
1201
1202
1203
1204
1205
```

```
                                .SBTTL  GLOBAL TEXT SECTION

                                ;%%%
```

```
                        .SBTTL  GLOBAL SUBROUTINES

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



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

                        .MACRO  ERROR,XYX
                                MOV     R4,$BDDAT
                                MOV     R2,$GDADR
                                MOV     MR0,$BDADR
                                ERRDF   XYX',EM'XYX',ERR'XYX'
                        .ENDM
                        .MACRO  ED$CALL XY
                                .LIST
                                ;******************** TEST 'XY' **********************
                                .NLIST
                                .ENDM
                                .MACRO  BADHEAD
                                .RADIX  10
                                ED$CALL \T$TESTNUM+1
                                .RADIX  8
                                .ENDM
                                .MACRO  K4ONLY  ?N2
                                CMP     MEMSZ,#2000
                                BNE     N2
                                EXIT    TST
                        N2:
                                .ENDM
                        .MACRO  MYINT
                                JSR     R5,DEBUG        ;WILL PUT CSR ADDR. INTO R1
                        .ENDM

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

                        .MACRO  MSTCLR
                        JSR     R5,..MSTCLR             ;CLEAR M8200,4,6,7
                        .ENDM

003162                  .MSTCLR:
003162  105077  177520          CLRB    @KMCSRH         ;*; CLEAR RUN, IF UP
003166  112777  000100  177512  MOVB    #BIT6,@KMCSRH   ;SET INST.
003174  142777  000300  177504  BICB    #BIT6!BIT7,@KMCSRH
003202  000205                 RTS     R5

                        ;
                        ;DEBUG   THIS ROUTINE IS ENTERED AT THE BEGINING OF EACH TEST
                        ;       IN ORDER TO LOAD THE CSR ADDR. INTO R1..
                        ;       ALSO THIS PROGRAM TRACKS ITSELF HERE.
```

```
                                        ;       CALL=JSR        R5,DEBUG
                                        ;
1298  003204                          DEBUG:
1299  003204  010537  003224                  MOV     R5,TESTAD       ;SAVE ADDR. OF TEST.
1300  003210  013701  002704                  MOV     KMCSR,R1        ;LOAD KMCSR INTP R1.
1301  003214  000240                          NOP
1302  003216  000240                          NOP
1303  003220  000240                          NOP
1304  003222  000205                          RTS R5
1305  003224  000000                  TESTAD: .WORD   0               ;LAST TEST ADDR.
1306  003226  000024                          .BLKW   20.             ;PATCH AREA.
1307
1308
1309
1310  003276                          ENDBUG:
1311                                  ;       UNSAFE TO PATCH ANY OTHER AREA.
1312
1313
1314
1315  003276                          .ROMCLK:
1316  003276  152777  000002  177402          BISB    #BIT1,@KMCSRH
1317  003304  012577  177404                  MOV     (R5)+,@KMP06
1318  003310  152777  000003  177370          BISB    #BIT1!BIT0,@KMCSRH
1319  003316  142777  000007  177362          BICB    #BIT2!BIT1!BIT0,@KMCSRH
1320  003324  000205                          RTS     R5
1321
1322  003326                          CLRALL:
1323                                          ;CLEARS C & Z BITS AND BR
1324  003326                                  ROMCLK
1325  003332  000400                          400                     ;0 TO BR
1326  003334                                  ROMCLK
1327  003340  063220                          63220                   ;SP(0) TO BR
1328  003342                                  ROMCLK
1329  003346  060400                          60400                   ;BR,SP(0) + BR
1330  003350  000207                          RTS     PC
1331
1332  003352                          SETBR0:
1333                                          ;SETS BR0 BIT
1334  003352                                  ROMCLK
1335  003356  000401                          401                     ;1 TO BR
1336  003360  000207                          RTS     PC
1337
1338  003362                          SETBR1:
1339                                          ;THIS SUBROUTINE SETS BR1 BIT
1340
1341  003362                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION
1342  003366  000402                          000402                  ;BR_002
1343  003370  000207                          RTS     PC
1344
1345  003372                          SETBR4:
1346                                          ;THIS SUBROUTINE SETS BR4 BIT
1347
1348  003372                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION
1349  003376  000420                          420
1350  003400  000207                          RTS     PC
1351
```

```
1352  003402                          SETBR7:
1353                                          ;THIS SUBROUTINE SETS BR7 BIT
1354
1355  003402                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION
1356  003406  000600                         600
1357  003410  000207                         RTS     PC
1358
1359
1360  003412                          SETZ:
1361                                          ;THIS SUBROUTINE SETS THE Z BIT
1362
1363  003412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION
1364  003416  000777                         000777                  ;BR_377
1365  003420  000207                         RTS     PC
1366
1367  003422                          RAMDAT:
1368                                          ;THIS SUBROUTINE LOADS R4 WITH THE LOWEST
1369                                          ;8 BITS OF THE CRAM PC.
1370
1371  003422  017605  000000                  MOV     @(SP),R5        ;GOOD DATA
1372  003426  062716  000002                  ADD     #2,(SP)         ;ADJUST STACK
1373  003432  005011                          CLR     (R1)            ;CLEAR BIT10
1374  003434  052711  000400                  BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION IN CRAM THAT
1375                                                                  ;JUMPED TO, IT LOADS BR WITH IT
1376  003440  005011                          CLR     (R1)            ;CLR BIT8
1377  003442                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION
1378  003446  061225                         061225                  ;MOV BR TO PORT 5
1379  003450  116104  000005                 MOVB    5(R1),R4        ;PUT "FOUND" IN R4
1380  003454  000207                         RTS     PC              ;RETURN
1381
1382
1383  003456                          MEMSET:
1384                                          ;THIS SUBROUTINE LOADS CRAM WITH SPECIAL INSTRUCTIONS
1385                                          ;FOR THE CRAM JUMP TEST. ALL CRAM LOCATIONS ARE LOADED
1386                                          ;WITH INSTRUCTIONS THAT MOVE A 37 TO THE BR, EXCEPT THE
1387                                          ;FOLLOWING CRAM ADDRESSES: 0,1,4,7,525,1777. THESE LOCATIONS
1388                                          ;CONTAIN INSTRUCTIONS WHICH LOAD THE BR WITH THE LOWEST
1389                                          ;8 BITS OF THAT CRAM ADDRESS.
1390
1391  003456  005000                          CLR     R0              ;R0 = CRAM ADDRESS
1392  003460  012711  002000          1$:     MOV     #BIT10,(R1)     ;SET ROMO
1393  003464  010061  000004                  MOV     R0,4(R1)        ;LOAD CRAM ADDRESS
1394  003470  012761  000437  000006         MOV     #437,6(R1)      ;LOAD INSTRUCTION
1395  003476  052711  020000                  BIS     #BIT13,(R1)     ;WRITE INSTRUCTION IN CRAM
1396  003502  005200                          INC     R0              ;NEXT ADDRESS
1397  003504  022700  002000                  CMP     #2000,R0        ;DONE YET?
1398  003510  001363                          BNE     1$              ;BR IF NO
1399  003512  005000                          CLR     R0              ;INDEX REGISTER
1400  003514  012711  002000          2$:     MOV     #BIT10,(R1)     ;SET ROMO
1401  003520  016061  003554  000004         MOV     CRAMA(R0),4(R1) ;LOAD CRAM ADDRESS IN SEL4
1402  003526  016061  003570  000006         MOV     INSTU(R0),6(R1) ;LOAD INSTRUCTION TO BE WRITTEN
1403  003534  052711  020000                  BIS     #BIT13,(R1)     ;WRITE CRAM!
1404  003540  005720                          TST     (R0)+           ;NEXT
1405  003542  022700  000014                  CMP     #14,R0          ;DONE YET?
1406  003546  001362                          BNE     2$              ;BR IF NO
1407  003550  005011                          CLR     (R1)            ;CLEAR ALL BITS
```

```
1408  003552  000207                          RTS     PC              ;RETURN
1409
1410  003554  000000  000001  000004  CRAMA:  .WORD   0,1,4,7,1777,525
1411  003562  000007  001777  000525
1412
1413  003570  000400                  INSTU:  000400          ;BR_0
1414  003572  000401                          000401          ;BR_1
1415  003574  000404                          000404          ;BR_4
1416  003576  000407                          000407          ;BR_7
1417  003600  000777                          000777          ;BR_377
1418  003602  000525                          000525          ;BR_125
1419
1420  003604                          SETVEC:
1421                                          ;THIS SUBROUTINE LOADS THE VECTORS AND VECTOR LEVELS
1422
1423  003604  012577  177064                  MOV     (R5)+,@KMRVEC   ;LOAD BASE VECTOR
1424  003610  012577  177064                  MOV     (R5)+,@KMTVEC   ;LOAD VECTOR + 2
1425  003614  012577  177056                  MOV     (R5)+,@KMRLVL   ;LOAD VECTOR + 4
1426  003620  012577  177056                  MOV     (R5)+,@KMTLVL   ;LOAD VECTOR + 6
1427  003624  000205                          RTS     R5              ;RETURN
1428
1429
1430  003626                          NPRSET:
1431                                          ;THIS SUBROUTINE LOADS IBUS REGISTERS 0-7
1432                                          ;WITH NPR INFORMATION (INBA, OUTBA, OUT DATA)
1433
1434  003626  010246                          MOV     R2,-(SP)        ;SAVE R2
1435  003630  005002                          CLR     R2              ;START AT IBUS REG 0
1436  003632  112561  000004          1$:     MOVB    (R5)+,4(R1)     ;LOAD PORT4
1437  003636  042737  000017  003654          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
1438  003644  050237  003654                  BIS     R2,2$           ;ADD ADDRESS TO INSTRUCTION
1439  003650                                  ROMCLK
1440  003654  122100                  2$:     122100                  ;MOVE PORT4 TO IBUS REG
1441  003656  005202                          INC     R2              ;NEXT ADDRESS
1442  003660  022702  000010                  CMP     #10,R2          ;ALL DONE?
1443  003664  001362                          BNE     1$              ;BR IF NO
1444  003666  012602                          MOV     (SP)+,R2        ;RESTORE R2
1445  003670  000205                          RTS     R5              ;RETURN
1446
1447
1448  003672                          MEMLD:
1449                                          ;THIS SUBROUTINE LOADS THE FIRST 8 LOCATIONS OF MAIN
1450                                          ;MEMORY WITH THIS DATA: 0,-1,,0,-1,125,252,125,252
1451
1452  003672  013637  002522                  MOV     @(SP)+,$TMP0    ;PUT POINTER TO DATA IN R0
1453  003676  062746  000002                  ADD     #2,-(SP)        ;ADJUST STACK
1454
1455  003702  013700  002522          MEMLD2: MOV     $TMP0,R0        ;GET ADDR.
1456  003706  012704  000010                  MOV     #10,R4          ;DO 8 LOADS
1457  003712                                  ROMCLK
1458  003716  010000                          010000                  ;MAR < 0
1459  003720                                  ROMCLK                  ;CLR    MAR HI
1460  003724  004000                          004000
1461  003726  112077  176760          1$:     MOVB    (R0)+,@KMP04    ;LOAD PORT4
1462  003732                                  ROMCLK
1463  003736  136500                          136500                  ;MOV DATA TO MEM, AUTO INC MAR
```

```
1464  003740  005304                        DEC     R4              ;DECREMENT COUNT
1465  003742  001371                        BNE     1$              ;BR IF NOT DONE
1466
1467  003744                                ROMCLK                  ;LOAD MEM ADDR. 0
1468  003750  010000                        10000
1469  003752  012703  000010                MOV     #10,R3          ;CHECK 8. MEM LOCS.
1470  003756  013700  002522                MOV     $TMP0,R0
1471  003762                        2$:     ROMCLK                  ;READ FROM MEM,PUT INTO PORT 4
1472  003766  055224                        55224
1473
1474  003770  112037  002612                MOVB    (R0)+,$GDDAT    ;EXPECTED.
1475  003774  117704  176712                MOVB    @KMP04,R4       ;RECIEVED.
1476  004000  123704  002612                CMPB    $GDDAT,R4       ;OK?
1477  004004  001414                        BEQ     3$
1478  004006                                ERROR   36
1479  004024  104455                        TRAP    C$ERDF
1480  004026  000044                        .WORD   36
1481  004030  006107                        .WORD   EM36
1482  004032  012346                        .WORD   ERR36
1483  004034  000402                        BR      4$
1484  004036  005303                3$:     DEC     R3              ;CHECKED ALL?
1485  004040  001350                        BNE     2$              ;NO-DO NEXT ONE.
1486  004042                        4$:
1487  004042  000207                        RTS     PC              ;RETURN
1488
1489
1490  004044                        SPLD:
1491
1492                                        ;THIS SUBROUTINE LOADS THE FIRST 8 SCRATCH PAD
1493                                        ;LOCATIONS WITH: 0,0,-1,-1,125,125,252,252
1494  004044  013600                        MOV     @(SP)+,R0       ;PUT POINTER TO DATA IN R5
1495  004046  062746  000002                ADD     #2,-(SP)        ;ADJUST STACK
1496  004052  005004                        CLR     R4              ;START AT SP ADDRESS 0
1497  004054  112077  176632        1$:     MOVB    (R0)+,@KMP04    ;LOAD PORT4 WITH DATA
1498  004060  042737  000017  004076        BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
1499  004066  050437  004076                BIS     R4,2$           ;ADD ADDRESS TO INSTRUCTION
1500  004072                                ROMCLK
1501  004076  123100                2$:     123100                  ;MOVE DATA TO SP
1502  004100  005204                        INC     R4              ;INCREMENT COUNT
1503  004102  022704  000010                CMP     #10,R4          ;DONE YET?
1504  004106  001362                        BNE     1$              ;BR IF NO
1505  004110  000207                        RTS     PC              ;RETURN
1506
1507
1508  004112                        CLRC:
1509
1510                                        ;THIS SUBROUTINE CLEARS THE MICRO PROCESSOR C BIT
1511  004112                                ROMCLK
1512  004116  010000                        010000                  ;MAR_0
1513  004120                                ROMCLK
1514  004124  040400                        040400!<0*20>           ;CLEAR C BIT
1515  004126  000207                        RTS     PC              ;RETURN
1516
1517
1518  004130                        SETC:
1519
                                            ;THIS SUBROUTINE SETS THE MICRO PROCESSOR C BIT
```

```
1520
1521  004130                                  ROMCLK
1522  004134  010003                          010003                  ;MAR_3
1523  004136                                  ROMCLK
1524  004142  040403                          040403!<0*20>           ;SET C BIT
1525  004144  000207                          RTS     PC              ;RETURN
1526
1527
1528
1529  004146  012704  000050          PRDEL:  MOV     #40.,R4
1530  004152                          1$:     DELAY   250.
1531  004152  012727  000372                  MOV     #250.,(PC)+
1532  004156  000000                          .WORD   0
1533  004160  013727  002116                  MOV     L$DLY,(PC)+
1534  004164  000000                          .WORD   0
1535  004166  005367  177772                  DEC     -6(PC)
1536  004172  001375                          BNE     .-4
1537  004174  005367  177756                  DEC     -22(PC)
1538  004200  001367                          BNE     .-20
1539  004202  005304                          DEC     R4
1540  004204  001362                          BNE     1$                      ;DELAY 1 SECOND
1541  004206  000207                          RTS     PC
1542
```

```
                        .SBTTL  GLOBAL ERROR REPORT SECTION

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

004210  047045 052045 047045  FM1:    .ASCIZ  /%N%T%N/
004216     000
004217     045 022516 052501  FMX:    .ASCIZ  /%N%AUNIT = %02%A ; FAILING UNIT ADDRESS = %06%N/
004224  044516 020124 020075
004232  047445 022462 020101
004240  020073 040506 046111
004246  047111 020107 047125
004254  052111 040440 042104
004262  042522 051523 036440
004270  022440 033117 047045
004276     000
004277     045 031517 051445  TFM1:   .ASCIZ  /%03%S5%03%S5%03%S5%03%N2/
004304  022465 031517 051445
004312  022465 031517 051445
004320  022465 031517 047045
004326  000062
004330  047445 022466 031123  TFM2:   .ASCIZ  /%06%S2%06%N2/
004336  047445 022466 031116
004344     000
004345     045 031517 051445  TFM5:   .ASCIZ  /%03%S5%03%N2/
004352  022465 031517 047045
004360  000062
004362  047045 047445 022463  TFM27:  .ASCIZ  /%N%03%S5%03%S7%03%N2/
004370  032523 047445 022463
004376  033523 047445 022463
004404  031116    000

004407     045 030516 040445  FMSG:   .ASCIZ  /%N1%APROGRAM CLOCK = 75 MICROSECONDS%N1/
004414  051120 043517 040522
004422  020115 046103 041517
004430  020113 020075 032467
004436  046440 041511 047522
004444  042523 047503 042116
004452  022523 030516    000
004457     045 030516 040445  SMSG:   .ASCIZ  /%N1%APROGRAM CLOCK = 115 MILLISECONDS%N1/
004464  051120 043517 040522
004472  020115 046103 041517
004500  020113 020075 030461
004506  020065 044515 046114
004514  051511 041505 047117
004522  051504 047045 000061
004530  042522 044507 052123  EM1:    .ASCIZ  &REGISTER ADDRESS TEST&
004536  051105 040440 042104
004544  042522 051523 052040
```

```
1599  004552  051505  000124
1600  004556  041111  051525  020052  EM2:    .ASCIZ  &IBUS* REGISTER DUAL ADDRESSING TEST&
1601  004564  042522  044507  052123
1602  004572  051105  042040  040525
1603  004600  020114  042101  051104
1604  004606  051505  044523  043516
1605  004614  052040  051505  000124
1606  004622  041111  051525  051040  EM30:   .ASCIZ  ''IBUS REGISTER DUAL ADDRESSING TEST''
1607  004630  043505  051511  042524
1608  004636  020122  052504  046101
1609  004644  040440  042104  042522
1610  004652  051523  047111  020107
1611  004660  042524  052123     000
1612  004665     102  020122  042522  EM3:    .ASCIZ  /BR REGISTER DATA TEST/
1613  004672  044507  052123  051105
1614  004700  042040  052101  020101
1615  004706  042524  052123     000
1616  004713     123  051103  052101  EM4:    .ASCIZ  /SCRATCH PAD DATA TEST/
1617  004720  044103  050040  042101
1618  004726  042040  052101  020101
1619  004734  042524  052123     000
1620  004741     123  051103  052101  EM5:    .ASCIZ  /SCRATCH PAD DUAL ADDRESSING TEST/
1621  004746  044103  050040  042101
1622  004754  042040  040525  020114
1623  004762  042101  051104  051505
1624  004770  044523  043516  052040
1625  004776  051505  000124
1626  005002  040515  047111  046440  EM6:    .ASCIZ  /MAIN MEMORY DATA TEST/
1627  005010  046505  051117  020131
1628  005016  040504  040524  052040
1629  005024  051505  000124
1630  005030  040515  047111  046440  EM7:    .ASCIZ  /MAIN MEMORY DUAL ADDRESSING TEST/
1631  005036  046505  051117  020131
1632  005044  052504  046101  040440
1633  005052  042104  042522  051523
1634  005060  047111  020107  042524
1635  005066  052123     000
1636  005071     101  052125  020117  EM10:   .ASCIZ  /AUTO MARINC FUNCTION TEST/
1637  005076  040515  044522  041516
1638  005104  043040  047125  052103
1639  005112  047511  020116  042524
1640  005120  052123     000
1641  005123     116  051120  052040  EM11:   .ASCIZ  /NPR TEST/
1642  005130  051505  000124
1643  005134  052515  052114  050111  EM12:   .ASCIZ  /MULTIPLE NPR TEST/
1644  005142  042514  047040  051120
1645  005150  052040  051505  000124
1646  005156  047516  020116  054105  EM13:   .ASCIZ  /NON EX MEM FAILED/
1647  005164  046440  046505  043040
1648  005172  044501  042514  000104
1649  005200  051120  043517  040522  EM14:   .ASCIZ  /PROGRAM CLOCK TEST/
1650  005206  020115  046103  041517
1651  005214  020113  042524  052123
1652  005222     000
1653  005223     101  052514  043040  EM15:   .ASCIZ  /ALU FUNCTION WITH C BIT CLEAR TEST/
1654  005230  047125  052103  047511
```

```
 1655  005236  020116  044527  044124
 1656  005244  041440  041040  052111
 1657  005252  041440  042514  051101
 1658  005260  052040  051505  000124
 1659  005266  047520  042527  020122  EM16:   .ASCIZ  /POWER FAIL: BUS INIT WAS NOT BLOCKED/
 1660  005274  040506  046111  020072
 1661  005302  052502  020123  047111
 1662  005310  052111  053440  051501
 1663  005316  047040  052117  041040
 1664  005324  047514  045503  042105
 1665  005332     000
 1666  005333                          EM35:
 1667  005333     106  051117  042503  EM17:   .ASCIZ  /FORCE POWER FAIL ERROR/
 1668  005340  050040  053517  051105
 1669  005346  043040  044501  020114
 1670  005354  051105  047522  000122
 1671  005362  047516  051511  020105  EM20:   .ASCIZ  /NOISE TEST ON IBUS*,IBUS,SPAD,MEMORY/
 1672  005370  042524  052123  047440
 1673  005376  020116  041111  051525
 1674  005404  026052  041111  051525
 1675  005412  051454  040520  026104
 1676  005420  042515  047515  054522
 1677  005426     000
 1678  005427     101  052514  041440  EM21:   .ASCIZ  /ALU C BIT TEST FAILURE/
 1679  005434  041040  052111  052040
 1680  005442  051505  020124  040506
 1681  005450  046111  051125  000105
 1682  005456  044524  042515  047440  EM22:   .ASCIZ  /TIME OUT ERROR/
 1683  005464  052125  042440  051122
 1684  005472  051117     000
 1685  005475     101  052514  043040  EM23:   .ASCIZ  /ALU FUNCTION TEST WITH C BIT SET/
 1686  005502  047125  052103  047511
 1687  005510  020116  042524  052123
 1688  005516  053440  052111  020110
 1689  005524  020103  044502  020124
 1690  005532  042523  000124
 1691  005536  050125  020103  042523  EM24:   .ASCIZ  /UPC SEQUENCE ERROR/
 1692  005544  052521  047105  042503
 1693  005552  042440  051122  051117
 1694  005560     000
 1695  005561     125  020120  040506  EM31:   .ASCIZ  'UP FAILED TO INTERRUPT'
 1696  005566  046111  042105  052040
 1697  005574  020117  047111  042524
 1698  005602  051122  050125  000124
 1699  005610  050125  044440  052116  EM32:   .ASCIZ  'UP INTERRUPTED TO WRONG VECTOR'
 1700  005616  051105  052522  052120
 1701  005624  042105  052040  020117
 1702  005632  051127  047117  020107
 1703  005640  042526  052103  051117
 1704  005646     000
 1705  005647     125  042516  050130  EM33:   .ASCIZ  'UNEXPECTED INTERRUPT FROM UP'
 1706  005654  041505  042524  020104
 1707  005662  047111  042524  051122
 1708  005670  050125  020124  051106
 1709  005676  046517  052440  000120
 1710  005704  046101  020125  046106  EM34:   .ASCIZ  'ALU FLAG TEST'
```

```
1711   005712  043501  052040  051505
1712   005720  000124
1713   005722  042510  046114  051040  EM25:   .ASCIZ  /HELL RAISER TEST/
1714   005730  044501  042523  020122
1715   005736  042524  052123     000
1716   005743     115  044501  052116  EM26:   .ASCIZ  /MAINTANCE REGISTER ERROR/
1717   005750  047101  042503  051040
1718   005756  043505  051511  042524
1719   005764  020122  051105  047522
1720   005772  000122
1721   005774  041111  051525  020052  EM27:   .ASCIZ  'IBUS* WRITE/READ ERROR'
1722   006002  051127  052111  027505
1723   006010  042522  042101  042440
1724   006016  051122  051117     000
1725   006023     111  051516  051124  EM28:   .ASCIZ  /INSTRUCTION TEST FAILURE/
1726   006030  041525  044524  047117
1727   006036  052040  051505  020124
1728   006044  040506  046111  051125
1729   006052  000105
1730   006054  041111  051525  047457  EM29:   .ASCIZ  'IBUS/OBUS WRITE/READ ERROR'
1731   006062  052502  020123  051127
1732   006070  052111  027505  042522
1733   006076  042101  042440  051122
1734   006104  051117     000
1735   006107     111  050117  046440  EM36:   .ASCIZ  'IOP MAIN MEM. LOAD ERROR-RUN MCPU MEM. DIAG.'
1736   006114  044501  020116  042515
1737   006122  027115  046040  040517
1738   006130  020104  051105  047522
1739   006136  026522  052522  020116
1740   006144  041515  052520  046440
1741   006152  046505  020056  044504
1742   006160  043501  000056
1743   006164  042523  030114  024040  EM99:   .ASCIZ  /SELO (CSR) DID NOT CLEAR/
1744   006172  051503  024522  042040
1745   006200  042111  047040  052117
1746   006206  041440  042514  051101
1747   006214     000
```

```
1748  006215    000                       DH0:    .ASCIZ  //
1749  006216  047507  042117  020040      DH1:    .ASCIZ  /GOOD    BAD     UPC     REGISTER/
1750  006224  020040  040502  020104
1751  006232  020040  020040  050125
1752  006240  020103  020040  020040
1753  006246  042522  044507  052123
1754  006254  051105    000
1755  006257     107  047517  020104      DH2:    .ASCIZ  /GOOD    BAD/
1756  006264  020040  041040  042101
1757  006272     000
1758  006273     107  047517  020104      DH3:    .ASCIZ  /GOOD    BAD     UPC     ADDRESS/
1759  006300  020040  041040  042101
1760  006306  020040  020040  052440
1761  006314  041520  020040  020040
1762  006322  040440  042104  042522
1763  006330  051523    000
1764  006333     107  047517  020104      DH4:    .ASCIZ  /GOOD    BAD     FUNCTION/
1765  006340  020040  041040  042101
1766  006346  020040  020040  043040
1767  006354  047125  052103  047511
1768  006362  000116
1769  006364  042522  027107  042440      DH27:   .ASCIZ  /REG. EXPECTED FOUND/
1770  006372  050130  041505  042524
1771  006400  020104  047506  047125
1772  006406  000104
1773
1774
1775
1776
1777                                      .EVEN
```

```
1778
1779                    ;-----------------------------------------------------------------------
1780                    ; MACRO'S NEEDED TO REPORT ERRORS
1781                    ;-----------------------------------------------------------------------
1782                    .MACRO  MDT0
1783                    .ENDM
1784
1785                    .MACRO  MDT1
1786                            PRINTB #TFM1,$GDDAT,$BDDAT,$GDADR
1787                    .ENDM
1788
1789                    .MACRO MDT2
1790                            PRINTB #TFM2,$GDDAT,$BDDAT
1791                    .ENDM
1792                    .MACRO  MDT27
1793                            PRINTB #TFM27,MR0,$GDDAT,$BDDAT
1794                    .ENDM
1795
1796                    .MACRO  $MD,ERNB,ERHM,ERFM
1797                    .NLIST
1798                    ; ERNB = ERROR NUMBER
1799                    ; ERFM = FORMAT NUMBER
1800                    ; ERHM = HEADER NUMBER
1801                    .LIST
1802                            BGNMSG  ERR'ERNB'
1803                            PRINTB  #FMX,LOGDEV,KMCSR
1804                            PRINTB  #FM1,#DH'ERHM'
1805                            MDT'ERFM'
1806                            ENDMSG
                        .ENDM
```

```
006410                                  $MD     1,2,2
006410                          ERR1::
006410  013746  002704                  MOV     KMCSR,-(SP)
006414  013746  002524                  MOV     LOGDEV,-(SP)
006420  012746  004217                  MOV     #FMX,-(SP)
006424  012746  000003                  MOV     #3,-(SP)
006430  010600                          MOV     SP,R0
006432  104414                          TRAP    C$PNTB
006434  062706  000010                  ADD     #10,SP
006440  012746  006257                  MOV     #DH2,-(SP)
006444  012746  004210                  MOV     #FM1,-(SP)
006450  012746  000002                  MOV     #2,-(SP)
006454  010600                          MOV     SP,R0
006456  104414                          TRAP    C$PNTB
006460  062706  000006                  ADD     #6,SP
006464  013746  002614                  MOV     $BDDAT,-(SP)
006470  013746  002612                  MOV     $GDDAT,-(SP)
006474  012746  004330                  MOV     #TFM2,-(SP)
006500  012746  000003                  MOV     #3,-(SP)
006504  010600                          MOV     SP,R0
006506  104414                          TRAP    C$PNTB
006510  062706  000010                  ADD     #10,SP
006514                          L10002:
006514  104423                          TRAP    C$MSG
006516                                  $MD     2,2,2
006516                          ERR2::
006516  013746  002704                  MOV     KMCSR,-(SP)
006522  013746  002524                  MOV     LOGDEV,-(SP)
006526  012746  004217                  MOV     #FMX,-(SP)
006532  012746  000003                  MOV     #3,-(SP)
006536  010600                          MOV     SP,R0
006540  104414                          TRAP    C$PNTB
006542  062706  000010                  ADD     #10,SP
006546  012746  006257                  MOV     #DH2,-(SP)
006552  012746  004210                  MOV     #FM1,-(SP)
006556  012746  000002                  MOV     #2,-(SP)
006562  010600                          MOV     SP,R0
006564  104414                          TRAP    C$PNTB
006566  062706  000006                  ADD     #6,SP
006572  013746  002614                  MOV     $BDDAT,-(SP)
006576  013746  002612                  MOV     $GDDAT,-(SP)
006602  012746  004330                  MOV     #TFM2,-(SP)
006606  012746  000003                  MOV     #3,-(SP)
006612  010600                          MOV     SP,R0
006614  104414                          TRAP    C$PNTB
006616  062706  000010                  ADD     #10,SP
006622                          L10003:
006622  104423                          TRAP    C$MSG
006624                                  $MD     3,2,2
006624                          ERR3::
006624  013746  002704                  MOV     KMCSR,-(SP)
006630  013746  002524                  MOV     LOGDEV,-(SP)
006634  012746  004217                  MOV     #FMX,-(SP)
006640  012746  000003                  MOV     #3,-(SP)
006644  010600                          MOV     SP,R0
006646  104414                          TRAP    C$PNTB
```

```
 1863  006650  062706  000010                 ADD     #10,SP
 1864  006654  012746  006257                 MOV     #DH2,-(SP)
 1865  006660  012746  004210                 MOV     #FM1,-(SP)
 1866  006664  012746  000002                 MOV     #2,-(SP)
 1867  006670  010600                         MOV     SP,R0
 1868  006672  104414                         TRAP    C$PNTB
 1869  006674  062706  000006                 ADD     #6,SP
 1870  006700  013746  002614                 MOV     $BDDAT,-(SP)
 1871  006704  013746  002612                 MOV     $GDDAT,-(SP)
 1872  006710  012746  004330                 MOV     #TFM2,-(SP)
 1873  006714  012746  000003                 MOV     #3,-(SP)
 1874  006720  010600                         MOV     SP,R0
 1875  006722  104414                         TRAP    C$PNTB
 1876  006724  062706  000010                 ADD     #10,SP
 1877  006730                         L10004:
 1878  006730  104423                         TRAP    C$MSG
 1879  006732                                 $MD     4,1,1
 1880  006732                         ERR4::
 1881  006732  013746  002704                 MOV     KMCSR,-(SP)
 1882  006736  013746  002524                 MOV     LOGDEV,-(SP)
 1883  006742  012746  004217                 MOV     #FMX,-(SP)
 1884  006746  012746  000003                 MOV     #3,-(SP)
 1885  006752  010600                         MOV     SP,R0
 1886  006754  104414                         TRAP    C$PNTB
 1887  006756  062706  000010                 ADD     #10,SP
 1888  006762  012746  006216                 MOV     #DH1,-(SP)
 1889  006766  012746  004210                 MOV     #FM1,-(SP)
 1890  006772  012746  000002                 MOV     #2,-(SP)
 1891  006776  010600                         MOV     SP,R0
 1892  007000  104414                         TRAP    C$PNTB
 1893  007002  062706  000006                 ADD     #6,SP
 1894  007006  013746  002606                 MOV     $GDADR,-(SP)
 1895  007012  013746  002614                 MOV     $BDDAT,-(SP)
 1896  007016  013746  002612                 MOV     $GDDAT,-(SP)
 1897  007022  012746  004277                 MOV     #TFM1,-(SP)
 1898  007026  012746  000004                 MOV     #4,-(SP)
 1899  007032  010600                         MOV     SP,R0
 1900  007034  104414                         TRAP    C$PNTB
 1901  007036  062706  000012                 ADD     #12,SP
 1902  007042                         L10005:
 1903  007042  104423                         TRAP    C$MSG
 1904  007044                                 $MD     5,1,1
 1905  007044                         ERR5::
 1906  007044  013746  002704                 MOV     KMCSR,-(SP)
 1907  007050  013746  002524                 MOV     LOGDEV,-(SP)
 1908  007054  012746  004217                 MOV     #FMX,-(SP)
 1909  007060  012746  000003                 MOV     #3,-(SP)
 1910  007064  010600                         MOV     SP,R0
 1911  007066  104414                         TRAP    C$PNTB
 1912  007070  062706  000010                 ADD     #10,SP
 1913  007074  012746  006216                 MOV     #DH1,-(SP)
 1914  007100  012746  004210                 MOV     #FM1,-(SP)
 1915  007104  012746  000002                 MOV     #2,-(SP)
 1916  007110  010600                         MOV     SP,R0
 1917  007112  104414                         TRAP    C$PNTB
 1918  007114  062706  000006                 ADD     #6,SP
```

```
1919  007120  013746  002606          MOV     $GDADR,-(SP)
1920  007124  013746  002614          MOV     $BDDAT,-(SP)
1921  007130  013746  002612          MOV     $GDDAT,-(SP)
1922  007134  012746  004277          MOV     #TFM1,-(SP)
1923  007140  012746  000004          MOV     #4,-(SP)
1924  007144  010600                  MOV     SP,R0
1925  007146  104414                  TRAP    C$PNTB
1926  007150  062706  000012          ADD     #12,SP
1927  007154                  L10006:
1928  007154  104423                  TRAP    C$MSG
1929  007156                          $MD     6,3,1
1930  007156                  ERR6::
1931  007156  013746  002704          MOV     KMCSR,-(SP)
1932  007162  013746  002524          MOV     LOGDEV,-(SP)
1933  007166  012746  004217          MOV     #FMX,-(SP)
1934  007172  012746  000003          MOV     #3,-(SP)
1935  007176  010600                  MOV     SP,R0
1936  007200  104414                  TRAP    C$PNTB
1937  007202  062706  000010          ADD     #10,SP
1938  007206  012746  006273          MOV     #DH3,-(SP)
1939  007212  012746  004210          MOV     #FM1,-(SP)
1940  007216  012746  000002          MOV     #2,-(SP)
1941  007222  010600                  MOV     SP,R0
1942  007224  104414                  TRAP    C$PNTB
1943  007226  062706  000006          ADD     #6,SP
1944  007232  013746  002606          MOV     $GDADR,-(SP)
1945  007236  013746  002614          MOV     $BDDAT,-(SP)
1946  007242  013746  002612          MOV     $GDDAT,-(SP)
1947  007246  012746  004277          MOV     #TFM1,-(SP)
1948  007252  012746  000004          MOV     #4,-(SP)
1949  007256  010600                  MOV     SP,R0
1950  007260  104414                  TRAP    C$PNTB
1951  007262  062706  000012          ADD     #12,SP
1952  007266                  L10007:
1953  007266  104423                  TRAP    C$MSG
1954  007270                          $MD     7,3,1
1955  007270                  ERR7::
1956  007270  013746  002704          MOV     KMCSR,-(SP)
1957  007274  013746  002524          MOV     LOGDEV,-(SP)
1958  007300  012746  004217          MOV     #FMX,-(SP)
1959  007304  012746  000003          MOV     #3,-(SP)
1960  007310  010600                  MOV     SP,R0
1961  007312  104414                  TRAP    C$PNTB
1962  007314  062706  000010          ADD     #10,SP
1963  007320  012746  006273          MOV     #DH3,-(SP)
1964  007324  012746  004210          MOV     #FM1,-(SP)
1965  007330  012746  000002          MOV     #2,-(SP)
1966  007334  010600                  MOV     SP,R0
1967  007336  104414                  TRAP    C$PNTB
1968  007340  062706  000006          ADD     #6,SP
1969  007344  013746  002606          MOV     $GDADR,-(SP)
1970  007350  013746  002614          MOV     $BDDAT,-(SP)
1971  007354  013746  002612          MOV     $GDDAT,-(SP)
1972  007360  012746  004277          MOV     #TFM1,-(SP)
1973  007364  012746  000004          MOV     #4,-(SP)
1974  007370  010600                  MOV     SP,R0
```

```
 1975  007372  104414
 1976  007374  062706  000012          TRAP    C$PNTB
 1977  007400                          ADD     #12,SP
 1978  007400  104423            L10010:
 1979  007402                          TRAP    C$MSG
 1980  007402                          $MD     10,3,1
 1981  007402  013746  002704    ERR10::
 1982  007406  013746  002524          MOV     KMCSR,-(SP)
 1983  007412  012746  004217          MOV     LOGDEV,-(SP)
 1984  007416  012746  000003          MOV     #FMX,-(SP)
 1985  007422  010600                  MOV     #3,-(SP)
 1986  007424  104414                  MOV     SP,R0
 1987  007426  062706  000010          TRAP    C$PNTB
 1988  007432  012746  006273          ADD     #10,SP
 1989  007436  012746  004210          MOV     #DH3,-(SP)
 1990  007442  012746  000002          MOV     #FM1,-(SP)
 1991  007446  010600                  MOV     #2,-(SP)
 1992  007450  104414                  MOV     SP,R0
 1993  007452  062706  000006          TRAP    C$PNTB
 1994  007456  013746  002606          ADD     #6,SP
 1995  007462  013746  002614          MOV     $GDADR,-(SP)
 1996  007466  013746  002612          MOV     $BDDAT,-(SP)
 1997  007472  012746  004277          MOV     $GDDAT,-(SP)
 1998  007476  012746  000004          MOV     #TFM1,-(SP)
 1999  007502  010600                  MOV     #4,-(SP)
 2000  007504  104414                  MOV     SP,R0
 2001  007506  062706  000012          TRAP    C$PNTB
 2002  007512                          ADD     #12,SP
 2003  007512  104423            L10011:
 2004  007514                          TRAP    C$MSG
 2005  007514                          $MD     11,2,2
 2006  007514  013746  002704    ERR11::
 2007  007520  013746  002524          MOV     KMCSR,-(SP)
 2008  007524  012746  004217          MOV     LOGDEV,-(SP)
 2009  007530  012746  000003          MOV     #FMX,-(SP)
 2010  007534  010600                  MOV     #3,-(SP)
 2011  007536  104414                  MOV     SP,R0
 2012  007540  062706  000010          TRAP    C$PNTB
 2013  007544  012746  006257          ADD     #10,SP
 2014  007550  012746  004210          MOV     #DH2,-(SP)
 2015  007554  012746  000002          MOV     #FM1,-(SP)
 2016  007560  010600                  MOV     #2,-(SP)
 2017  007562  104414                  MOV     SP,R0
 2018  007564  062706  000006          TRAP    C$PNTB
 2019  007570  013746  002614          ADD     #6,SP
 2020  007574  013746  002612          MOV     $BDDAT,-(SP)
 2021  007600  012746  004330          MOV     $GDDAT,-(SP)
 2022  007604  012746  000003          MOV     #TFM2,-(SP)
 2023  007610  010600                  MOV     #3,-(SP)
 2024  007612  104414                  MOV     SP,R0
 2025  007614  062706  000010          TRAP    C$PNTB
 2026  007620                          ADD     #10,SP
 2027  007620  104423            L10012:
 2028  007622                          TRAP    C$MSG
 2029  007622                          $MD     12,2,2
 2030  007622  013746  002704    ERR12::
                                        MOV     KMCSR,-(SP)
```

```
2031  007626  013746  002524          MOV     LOGDEV,-(SP)
2032  007632  012746  004217          MOV     #FMX,-(SP)
2033  007636  012746  000003          MOV     #3,-(SP)
2034  007642  010600                  MOV     SP,R0
2035  007644  104414                  TRAP    C$PNTB
2036  007646  062706  000010          ADD     #10,SP
2037  007652  012746  006257          MOV     #DH2,-(SP)
2038  007656  012746  004210          MOV     #FM1,-(SP)
2039  007662  012746  000002          MOV     #2,-(SP)
2040  007666  010600                  MOV     SP,R0
2041  007670  104414                  TRAP    C$PNTB
2042  007672  062706  000006          ADD     #6,SP
2043  007676  013746  002614          MOV     $BDDAT,-(SP)
2044  007702  013746  002612          MOV     $GDDAT,-(SP)
2045  007706  012746  004330          MOV     #TFM2,-(SP)
2046  007712  012746  000003          MOV     #3,-(SP)
2047  007716  010600                  MOV     SP,R0
2048  007720  104414                  TRAP    C$PNTB
2049  007722  062706  000010          ADD     #10,SP
2050  007726                  L10013:
2051  007726  104423                  TRAP    C$MSG
2052  007730                          $MD     13,0,0
2053  007730                  ERR13::
2054  007730  013746  002704          MOV     KMCSR,-(SP)
2055  007734  013746  002524          MOV     LOGDEV,-(SP)
2056  007740  012746  004217          MOV     #FMX,-(SP)
2057  007744  012746  000003          MOV     #3,-(SP)
2058  007750  010600                  MOV     SP,R0
2059  007752  104414                  TRAP    C$PNTB
2060  007754  062706  000010          ADD     #10,SP
2061  007760  012746  006215          MOV     #DH0,-(SP)
2062  007764  012746  004210          MOV     #FM1,-(SP)
2063  007770  012746  000002          MOV     #2,-(SP)
2064  007774  010600                  MOV     SP,R0
2065  007776  104414                  TRAP    C$PNTB
2066  010000  062706  000006          ADD     #6,SP
2067  010004                  L10014:
2068  010004  104423                  TRAP    C$MSG
2069  010006                          $MD     14,2,2
2070  010006                  ERR14::
2071  010006  013746  002704          MOV     KMCSR,-(SP)
2072  010012  013746  002524          MOV     LOGDEV,-(SP)
2073  010016  012746  004217          MOV     #FMX,-(SP)
2074  010022  012746  000003          MOV     #3,-(SP)
2075  010026  010600                  MOV     SP,R0
2076  010030  104414                  TRAP    C$PNTB
2077  010032  062706  000010          ADD     #10,SP
2078  010036  012746  006257          MOV     #DH2,-(SP)
2079  010042  012746  004210          MOV     #FM1,-(SP)
2080  010046  012746  000002          MOV     #2,-(SP)
2081  010052  010600                  MOV     SP,R0
2082  010054  104414                  TRAP    C$PNTB
2083  010056  062706  000006          ADD     #6,SP
2084  010062  013746  002614          MOV     $BDDAT,-(SP)
2085  010066  013746  002612          MOV     $GDDAT,-(SP)
2086  010072  012746  004330          MOV     #TFM2,-(SP)
```

```
2087  010076  012746  000003          MOV     #3,-(SP)
2088  010102  010600                  MOV     SP,R0
2089  010104  104414                  TRAP    C$PNTB
2090  010106  062706  000010          ADD     #10,SP
2091  010112                  L10015:
2092  010112  104423                  TRAP    C$MSG
2093  010114                          $MD     15,4,27
2094  010114                  ERR15::
2095  010114  013746  002704          MOV     KMCSR,-(SP)
2096  010120  013746  002524          MOV     LOGDEV,-(SP)
2097  010124  012746  004217          MCV     #FMX,-(SP)
2098  010130  012746  000003          MOV     #3,-(SP)
2099  010134  010600                  MOV     SP,R0
2100  010136  104414                  TRAP    C$PNTB
2101  010140  062706  000010          ADD     #10,SP
2102  010144  012746  006333          MOV     #DH4,-(SP)
2103  010150  012746  004210          MOV     #FM1,-(SP)
2104  010154  012746  000002          MOV     #2,-(SP)
2105  010160  010600                  MOV     SP,R0
2106  010162  104414                  TRAP    C$PNTB
2107  010164  062706  000006          ADD     #6,SP
2108  010170  013746  002614          MOV     $BDDAT,-(SP)
2109  010174  013746  002612          MOV     $GDDAT,-(SP)
2110  010200  013746  002576          MOV     MR0,-(SP)
2111  010204  012746  004362          MOV     #TFM27,-(SP)
2112  010210  012746  000004          MOV     #4,-(SP)
2113  010214  010600                  MOV     SP,R0
2114  010216  104414                  TRAP    C$PNTB
2115  010220  062706  000012          ADD     #12,SP
2116  010224                  L10016:
2117  010224  104423                  TRAP    C$MSG
2118  010226                          $MD     16,0,0
2119  010226                  ERR16::
2120  010226  013746  002704          MOV     KMCSR,-(SP)
2121  010232  013746  002524          MOV     LOGDEV,-(SP)
2122  010236  012746  004217          MOV     #FMX,-(SP)
2123  010242  012746  000003          MOV     #3,-(SP)
2124  010246  010600                  MOV     SP,R0
2125  010250  104414                  TRAP    C$PNTB
2126  010252  062706  000010          ADD     #10,SP
2127  010256  012746  006215          MOV     #DH0,-(SP)
2128  010262  012746  004210          MOV     #FM1,-(SP)
2129  010266  012746  000002          MOV     #2,-(SP)
2130  010272  010600                  MOV     SP,R0
2131  010274  104414                  TRAP    C$PNTB
2132  010276  062706  000006          ADD     #6,SP
2133  010302                  L10017:
2134  010302  104423                  TRAP    C$MSG
2135  010304                          $MD     17,0,0
2136  010304                  ERR17::
2137  010304  013746  002704          MOV     KMCSR,-(SP)
2138  010310  013746  002524          MOV     LOGDEV,-(SP)
2139  010314  012746  004217          MOV     #FMX,-(SP)
2140  010320  012746  000003          MOV     #3,-(SP)
2141  010324  010600                  MOV     SP,R0
2142  010326  104414                  TRAP    C$PNTB
```

```
2143  010330  062706  000010          ADD     #10,SP
2144  010334  012746  006215          MOV     #DH0,-(SP)
2145  010340  012746  004210          MOV     #FM1,-(SP)
2146  010344  012746  000002          MOV     #2,-(SP)
2147  010350  010600                  MOV     SP,R0
2148  010352  104414                  TRAP    C$PNTB
2149  010354  062706  000006          ADD     #6,SP
2150  010360                  L10020:
2151  010360  104423                  TRAP    C$MSG
2152  010362                          $MD     20,2,2
2153  010362                  ERR20::
2154  010362  013746  002704          MOV     KMCSR,-(SP)
2155  010366  013746  002524          MOV     LOGDEV,-(SP)
2156  010372  012746  004217          MOV     #FMX,-(SP)
2157  010376  012746  000003          MOV     #3,-(SP)
2158  010402  010600                  MOV     SP,R0
2159  010404  104414                  TRAP    C$PNTB
2160  010406  062706  000010          ADD     #10,SP
2161  010412  012746  006257          MOV     #DH2,-(SP)
2162  010416  012746  004210          MOV     #FM1,-(SP)
2163  010422  012746  000002          MOV     #2,-(SP)
2164  010426  010600                  MOV     SP,R0
2165  010430  104414                  TRAP    C$PNTB
2166  010432  062706  000006          ADD     #6,SP
2167  010436  013746  002614          MOV     $BDDAT,-(SP)
2168  010442  013746  002612          MOV     $GDDAT,-(SP)
2169  010446  012746  004330          MOV     #TFM2,-(SP)
2170  010452  012746  000003          MOV     #3,-(SP)
2171  010456  010600                  MOV     SP,R0
2172  010460  104414                  TRAP    C$PNTB
2173  010462  062706  000010          ADD     #10,SP
2174  010466                  L10021:
2175  010466  104423                  TRAP    C$MSG
2176  010470                          $MD     21,0,0
2177  010470                  ERR21::
2178  010470  013746  002704          MOV     KMCSR,-(SP)
2179  010474  013746  002524          MOV     LOGDEV,-(SP)
2180  010500  012746  004217          MOV     #FMX,-(SP)
2181  010504  012746  000003          MOV     #3,-(SP)
2182  010510  010600                  MOV     SP,R0
2183  010512  104414                  TRAP    C$PNTB
2184  010514  062706  000010          ADD     #10,SP
2185  010520  012746  006215          MOV     #DH0,-(SP)
2186  010524  012746  004210          MOV     #FM1,-(SP)
2187  010530  012746  000002          MOV     #2,-(SP)
2188  010534  010600                  MOV     SP,R0
2189  010536  104414                  TRAP    C$PNTB
2190  010540  062706  000006          ADD     #6,SP
2191  010544                  L10022:
2192  010544  104423                  TRAP    C$MSG
2193  010546                          $MD     22,0,0
2194  010546                  ERR22::
2195  010546  013746  002704          MOV     KMCSR,-(SP)
2196  010552  013746  002524          MOV     LOGDEV,-(SP)
2197  010556  012746  004217          MOV     #FMX,-(SP)
2198  010562  012746  000003          MOV     #3,-(SP)
```

```
2199  010566  010600                        MOV     SP,R0
2200  010570  104414                        TRAP    C$PNTB
2201  010572  062706  000010                ADD     #10,SP
2202  010576  012746  006215                MOV     #DH0,-(SP)
2203  010602  012746  004210                MOV     #FM1,-(SP)
2204  010606  012746  000002                MOV     #2,-(SP)
2205  010612  010600                        MOV     SP,R0
2206  010614  104414                        TRAP    C$PNTB
2207  010616  062706  000006                ADD     #6,SP
2208  010622                        L10023:
2209  010622  104423                        TRAP    C$MSG
2210  010624                                $MD     23,4,1
2211  010624                        ERR23::
2212  010624  013746  002704                MOV     KMCSR,-(SP)
2213  010630  013746  002524                MOV     LOGDEV,-(SP)
2214  010634  012746  004217                MOV     #FMX,-(SP)
2215  010640  012746  000003                MOV     #3,-(SP)
2216  010644  010600                        MOV     SP,R0
2217  010646  104414                        TRAP    C$PNTB
2218  010650  062706  000010                ADD     #10,SP
2219  010654  012746  006333                MOV     #DH4,-(SP)
2220  010660  012746  004210                MOV     #FM1,-(SP)
2221  010664  012746  000002                MOV     #2,-(SP)
2222  010670  010600                        MOV     SP,R0
2223  010672  104414                        TRAP    C$PNTB
2224  010674  062706  000006                ADD     #6,SP
2225  010700  013746  002606                MOV     $GDADR,-(SP)
2226  010704  013746  002614                MOV     $BDDAT,-(SP)
2227  010710  013746  002612                MOV     $GDDAT,-(SP)
2228  010714  012746  004277                MOV     #TFM1,-(SP)
2229  010720  012746  000004                MOV     #4,-(SP)
2230  010724  010600                        MOV     SP,R0
2231  010726  104414                        TRAP    C$PNTB
2232  010730  062706  000012                ADD     #12,SP
2233  010734                        L10024:
2234  010734  104423                        TRAP    C$MSG
2235  010736                                $MD     24,0,0
2236  010736                        ERR24::
2237  010736  013746  002704                MOV     KMCSR,-(SP)
2238  010742  013746  002524                MOV     LOGDEV,-(SP)
2239  010746  012746  004217                MOV     #FMX,-(SP)
2240  010752  012746  000003                MOV     #3,-(SP)
2241  010756  010600                        MOV     SP,R0
2242  010760  104414                        TRAP    C$PNTB
2243  010762  062706  000010                ADD     #10,SP
2244  010766  012746  006215                MOV     #DH0,-(SP)
2245  010772  012746  004210                MOV     #FM1,-(SP)
2246  010776  012746  000002                MOV     #2,-(SP)
2247  011002  010600                        MOV     SP,R0
2248  011004  104414                        TRAP    C$PNTB
2249  011006  062706  000006                ADD     #6,SP
2250  011012                        L10025:
2251  011012  104423                        TRAP    C$MSG
2252  011014                                $MD     25,2,2
2253  011014                        ERR25::
2254  011014  013746  002704                MOV     KMCSR,-(SP)
```

```
2255  011020  013746  002524                  MOV     LOGDEV,-(SP)
2256  011024  012746  004217                  MOV     #FMX,-(SP)
2257  011030  012746  000003                  MOV     #3,-(SP)
2258  011034  010600                          MOV     SP,R0
2259  011036  104414                          TRAP    C$PNTB
2260  011040  062706  000010                  ADD     #10,SP
2261  011044  012746  006257                  MOV     #DH2,-(SP)
2262  011050  012746  004210                  MOV     #FM1,-(SP)
2263  011054  012746  000002                  MOV     #2,-(SP)
2264  011060  010600                          MOV     SP,R0
2265  011062  104414                          TRAP    C$PNTB
2266  011064  062706  000006                  ADD     #6,SP
2267  011070  013746  002614                  MOV     $BDDAT,-(SP)
2268  011074  013746  002612                  MOV     $GDDAT,-(SP)
2269  011100  012746  004330                  MOV     #TFM2,-(SP)
2270  011104  012746  000003                  MOV     #3,-(SP)
2271  011110  010600                          MOV     SP,R0
2272  011112  104414                          TRAP    C$PNTB
2273  011114  062706  000010                  ADD     #10,SP
2274  011120                          L10026:
2275  011120  104423                          TRAP    C$MSG
2276  011122                          $MD     26,2,2
2277  011122                          ERR26::
2278  011122  013746  002704                  MOV     KMCSR,-(SP)
2279  011126  013746  002524                  MOV     LOGDEV,-(SP)
2280  011132  012746  004217                  MOV     #FMX,-(SP)
2281  011136  012746  000003                  MOV     #3,-(SP)
2282  011142  010600                          MOV     SP,R0
2283  011144  104414                          TRAP    C$PNTB
2284  011146  062706  000010                  ADD     #10,SP
2285  011152  012746  006257                  MOV     #DH2,-(SP)
2286  011156  012746  004210                  MOV     #FM1,-(SP)
2287  011162  012746  000002                  MOV     #2,-(SP)
2288  011166  010600                          MOV     SP,R0
2289  011170  104414                          TRAP    C$PNTB
2290  011172  062706  000006                  ADD     #6,SP
2291  011176  013746  002614                  MOV     $BDDAT,-(SP)
2292  011202  013746  002612                  MOV     $GDDAT,-(SP)
2293  011206  012746  004330                  MOV     #TFM2,-(SP)
2294  011212  012746  000003                  MOV     #3,-(SP)
2295  011216  010600                          MOV     SP,R0
2296  011220  104414                          TRAP    C$PNTB
2297  011222  062706  000010                  ADD     #10,SP
2298  011226                          L10027:
2299  011226  104423                          TRAP    C$MSG
2300  011230                          $MD     27,27,27
2301  011230                          ERR27::
2302  011230  013746  002704                  MOV     KMCSR,-(SP)
2303  011234  013746  002524                  MOV     LOGDEV,-(SP)
2304  011240  012746  004217                  MOV     #FMX,-(SP)
2305  011244  012746  000003                  MOV     #3,-(SP)
2306  011250  010600                          MOV     SP,R0
2307  011252  104414                          TRAP    C$PNTB
2308  011254  062706  000010                  ADD     #10,SP
2309  011260  012746  006364                  MOV     #DH27,-(SP)
2310  011264  012746  004210                  MOV     #FM1,-(SP)
```

```
 2311  011270  012746  000002          MOV     #2,-(SP)
 2312  011274  010600                  MOV     SP,R0
 2313  011276  104414                  TRAP    C$PNTB
 2314  011300  062706  000006          ADD     #6,SP
 2315  011304  013746  002614          MOV     $BDDAT,-(SP)
 2316  011310  013746  002612          MOV     $GDDAT,-(SP)
 2317  011314  013746  002576          MOV     MR0,-(SP)
 2318  011320  012746  004362          MOV     #TFM27,-(SP)
 2319  011324  012746  000004          MOV     #4,-(SP)
 2320  011330  010600                  MOV     SP,R0
 2321  011332  104414                  TRAP    C$PNTB
 2322  011334  062706  000012          ADD     #12,SP
 2323  011340                  L10030:
 2324  011340  104423                  TRAP    C$MSG
 2325  011342                          $MD     28,2,2
 2326  011342                  ERR28::
 2327  011342  013746  002704          MOV     KMCSR,-(SP)
 2328  011346  013746  002524          MOV     LOGDEV,-(SP)
 2329  011352  012746  004217          MOV     #FMX,-(SP)
 2330  011356  012746  000003          MOV     #3,-(SP)
 2331  011362  010600                  MOV     SP,R0
 2332  011364  104414                  TRAP    C$PNTB
 2333  011366  062706  000010          ADD     #10,SP
 2334  011372  012746  006257          MOV     #DH2,-(SP)
 2335  011376  012746  004210          MOV     #FM1,-(SP)
 2336  011402  012746  000002          MOV     #2,-(SP)
 2337  011406  010600                  MOV     SP,R0
 2338  011410  104414                  TRAP    C$PNTB
 2339  011412  062706  000006          ADD     #6,SP
 2340  011416  013746  002614          MOV     $BDDAT,-(SP)
 2341  011422  013746  002612          MOV     $GDDAT,-(SP)
 2342  011426  012746  004330          MOV     #TFM2,-(SP)
 2343  011432  012746  000003          MOV     #3,-(SP)
 2344  011436  010600                  MOV     SP,R0
 2345  011440  104414                  TRAP    C$PNTB
 2346  011442  062706  000010          ADD     #10,SP
 2347  011446                  L10031:
 2348  011446  104423                  TRAP    C$MSG
 2349  011450                          $MD     29,27,27
 2350  011450                  ERR29::
 2351  011450  013746  002704          MOV     KMCSR,-(SP)
 2352  011454  013746  002524          MOV     LOGDEV,-(SP)
 2353  011460  012746  004217          MOV     #FMX,-(SP)
 2354  011464  012746  000003          MOV     #3,-(SP)
 2355  011470  010600                  MOV     SP,R0
 2356  011472  104414                  TRAP    C$PNTB
 2357  011474  062706  000010          ADD     #10,SP
 2358  011500  012746  006364          MOV     #DH27,-(SP)
 2359  011504  012746  004210          MOV     #FM1,-(SP)
 2360  011510  012746  000002          MOV     #2,-(SP)
 2361  011514  010600                  MOV     SP,R0
 2362  011516  104414                  TRAP    C$PNTB
 2363  011520  062706  000006          ADD     #6,SP
 2364  011524  013746  002614          MOV     $BDDAT,-(SP)
 2365  011530  013746  002612          MOV     $GDDAT,-(SP)
 2366  011534  013746  002576          MOV     MR0,-(SP)
```

```
 2367  011540  012746  004362          MOV     #TFM27,-(SP)
 2368  011544  012746  000004          MOV     #4,-(SP)
 2369  011550  010600                  MOV     SP,R0
 2370  011552  104414                  TRAP    C$PNTB
 2371  011554  062706  000012          ADD     #12,SP
 2372  011560                  L10032:
 2373  011560  104423                  TRAP    C$MSG
 2374  011562                          $MD     30,2,2
 2375  011562                  ERR30::
 2376  011562  013746  002704          MOV     KMCSR,-(SP)
 2377  011566  013746  002524          MOV     LOGDEV,-(SP)
 2378  011572  012746  004217          MOV     #FMX,-(SP)
 2379  011576  012746  000003          MOV     #3,-(SP)
 2380  011602  010600                  MOV     SP,R0
 2381  011604  104414                  TRAP    C$PNTB
 2382  011606  062706  000010          ADD     #10,SP
 2383  011612  012746  006257          MOV     #DH2,-(SP)
 2384  011616  012746  004210          MOV     #FM1,-(SP)
 2385  011622  012746  000002          MOV     #2,-(SP)
 2386  011626  010600                  MOV     SP,R0
 2387  011630  104414                  TRAP    C$PNTB
 2388  011632  062706  000006          ADD     #6,SP
 2389  011636  013746  002614          MOV     $BDDAT,-(SP)
 2390  011642  013746  002612          MOV     $GDDAT,-(SP)
 2391  011646  012746  004330          MOV     #TFM2,-(SP)
 2392  011652  012746  000003          MOV     #3,-(SP)
 2393  011656  010600                  MOV     SP,R0
 2394  011660  104414                  TRAP    C$PNTB
 2395  011662  062706  000010          ADD     #10,SP
 2396  011666                  L10033:
 2397  011666  104423                  TRAP    C$MSG
 2398  011670                          $MD     31,0,0
 2399  011670                  ERR31::
 2400  011670  013746  002704          MOV     KMCSR,-(SP)
 2401  011674  013746  002524          MOV     LOGDEV,-(SP)
 2402  011700  012746  004217          MOV     #FMX,-(SP)
 2403  011704  012746  000003          MOV     #3,-(SP)
 2404  011710  010600                  MOV     SP,R0
 2405  011712  104414                  TRAP    C$PNTB
 2406  011714  062706  000010          ADD     #10,SP
 2407  011720  012746  006215          MOV     #DH0,-(SP)
 2408  011724  012746  004210          MOV     #FM1,-(SP)
 2409  011730  012746  000002          MOV     #2,-(SP)
 2410  011734  010600                  MOV     SP,R0
 2411  011736  104414                  TRAP    C$PNTB
 2412  011740  062706  000006          ADD     #6,SP
 2413  011744                  L10034:
 2414  011744  104423                  TRAP    C$MSG
 2415  011746                          $MD     32,0,0
 2416  011746                  ERR32::
 2417  011746  013746  002704          MOV     KMCSR,-(SP)
 2418  011752  013746  002524          MOV     LOGDEV,-(SP)
 2419  011756  012746  004217          MOV     #FMX,-(SP)
 2420  011762  012746  000003          MOV     #3,-(SP)
 2421  011766  010600                  MOV     SP,R0
 2422  011770  104414                  TRAP    C$PNTB
```

```
2423  011772  062706  000010                ADD     #10,SP
2424  011776  012746  006215                MOV     #DH0,-(SP)
2425  012002  012746  004210                MOV     #FM1,-(SP)
2426  012006  012746  000002                MOV     #2,-(SP)
2427  012012  010600                         MOV     SP,R0
2428  012014  104414                         TRAP    C$PNTB
2429  012016  062706  000006                ADD     #6,SP
2430  012022                        L10035:
2431  012022  104423                         TRAP    C$MSG
2432  012024                                  $MD     33,2,2
2433  012024                        ERR33::
2434  012024  013746  002704                MOV     KMCSR,-(SP)
2435  012030  013746  002524                MOV     LOGDEV,-(SP)
2436  012034  012746  004217                MOV     #FMX,-(SP)
2437  012040  012746  000003                MOV     #3,-(SP)
2438  012044  010600                         MOV     SP,R0
2439  012046  104414                         TRAP    C$PNTB
2440  012050  062706  000010                ADD     #10,SP
2441  012054  012746  006257                MOV     #DH2,-(SP)
2442  012060  012746  004210                MOV     #FM1,-(SP)
2443  012064  012746  000002                MOV     #2,-(SP)
2444  012070  010600                         MOV     SP,R0
2445  012072  104414                         TRAP    C$PNTB
2446  012074  062706  000006                ADD     #6,SP
2447  012100  013746  002614                MOV     $BDDAT,-(SP)
2448  012104  013746  002612                MOV     $GDDAT,-(SP)
2449  012110  012746  004330                MOV     #TFM2,-(SP)
2450  012114  012746  000003                MOV     #3,-(SP)
2451  012120  010600                         MOV     SP,R0
2452  012122  104414                         TRAP    C$PNTB
2453  012124  062706  000010                ADD     #10,SP
2454  012130                        L10036:
2455  012130  104423                         TRAP    C$MSG
2456  012132                                  $MD     34,2,2
2457  012132                        ERR34::
2458  012132  013746  002704                MOV     KMCSR,-(SP)
2459  012136  013746  002524                MOV     LOGDEV,-(SP)
2460  012142  012746  004217                MOV     #FMX,-(SP)
2461  012146  012746  000003                MOV     #3,-(SP)
2462  012152  010600                         MOV     SP,R0
2463  012154  104414                         TRAP    C$PNTB
2464  012156  062706  000010                ADD     #10,SP
2465  012162  012746  006257                MOV     #DH2,-(SP)
2466  012166  012746  004210                MOV     #FM1,-(SP)
2467  012172  012746  000002                MOV     #2,-(SP)
2468  012176  010600                         MOV     SP,R0
2469  012200  104414                         TRAP    C$PNTB
2470  012202  062706  000006                ADD     #6,SP
2471  012206  013746  002614                MOV     $BDDAT,-(SP)
2472  012212  013746  002612                MOV     $GDDAT,-(SP)
2473  012216  012746  004330                MOV     #TFM2,-(SP)
2474  012222  012746  000003                MOV     #3,-(SP)
2475  012226  010600                         MOV     SP,R0
2476  012230  104414                         TRAP    C$PNTB
2477  012232  062706  000010                ADD     #10,SP
2478  012236                        L10037:
```

```
2479  012236  104423                        TRAP    C$MSG
2480  012240                                $MD     35,2,2
2481  012240                        ERR35::
2482  012240  013746  002704                MOV     KMCSR,-(SP)
2483  012244  013746  002524                MOV     LOGDEV,-(SP)
2484  012250  012746  004217                MOV     #FMX,-(SP)
2485  012254  012746  000003                MOV     #3,-(SP)
2486  012260  010600                        MOV     SP,R0
2487  012262  104414                        TRAP    C$PNTB
2488  012264  062706  000010                ADD     #10,SP
2489  012270  012746  006257                MOV     #DH2,-(SP)
2490  012274  012746  004210                MOV     #FM1,-(SP)
2491  012300  012746  000002                MOV     #2,-(SP)
2492  012304  010600                        MOV     SP,R0
2493  012306  104414                        TRAP    C$PNTB
2494  012310  062706  000006                ADD     #6,SP
2495  012314  013746  002614                MOV     $BDDAT,-(SP)
2496  012320  013746  002612                MOV     $GDDAT,-(SP)
2497  012324  012746  004330                MOV     #TFM2,-(SP)
2498  012330  012746  000003                MOV     #3,-(SP)
2499  012334  010600                        MOV     SP,R0
2500  012336  104414                        TRAP    C$PN1B
2501  012340  062706  000010                ADD     #10,SP
2502  012344                        L10040:
2503  012344  104423                        TRAP    C$MSG
2504  012346                                $MD     36,2,2
2505  012346                        ERR36::
2506  012346  013746  002704                MOV     KMCSR,-(SP)
2507  012352  013746  002524                MOV     LOGDEV,-(SP)
2508  012356  012746  004217                MOV     #FMX,-(SP)
2509  012362  012746  000003                MOV     #3,-(SP)
2510  012366  010600                        MOV     SP,R0
2511  012370  104414                        TRAP    C$PNTB
2512  012372  062706  000010                ADD     #10,SP
2513  012376  012746  006257                MOV     #DH2,-(SP)
2514  012402  012746  004210                MOV     #FM1,-(SP)
2515  012406  012746  000002                MOV     #2,-(SP)
2516  012412  010600                        MOV     SP,R0
2517  012414  104414                        TRAP    C$PNTB
2518  012416  062706  000006                ADD     #6,SP
2519  012422  013746  002614                MOV     $BDDAT,-(SP)
2520  012426  013746  002612                MOV     $GDDAT,-(SP)
2521  012432  012746  004330                MOV     #TFM2,-(SP)
2522  012436  012746  000003                MOV     #3,-(SP)
2523  012442  010600                        MOV     SP,R0
2524  012444  104414                        TRAP    C$PNTB
2525  012446  062706  000010                ADD     #10,SP
2526  012452                        L10041:
2527  012452  104423                        TRAP    C$MSG
```

```
                                .SBTTL  REPORT CODING SECTION


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

012454                                  BGNRPT
012454                          L$RPT::

012454                                  EXIT    RPT
012454  000167                          .WORD   J$JMP
012456  000000                          .WORD   L10042-2-.

012460                                  ENDRPT
012460                          L10042:
012460  104425                          TRAP    C$RPT

012462                                  BGNPROT
012462                          L$PROT::
012462  177777                          -1
012464  177777                          -1
012466  177777                          -1
012470                                  ENDPROT

012470                                  BGNAUTO
012470                          L$AUTO::
012470                                  ENDAUTO
012470                          L10044:
012470  104461                          TRAP    C$AUTO
```

```
                                .SBTTL  INITIALIZE SECTION

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

012472                                  BGNINIT
012472                          L$INIT::

                                ;INITIALIZE SUBROUTINE STACK
012472  012705  003116                  MOV     #SSTACK,R5
                                ;STORE BASE LEVEL PROGRAM STACK POINTER
012476  010637  002526                  MOV     SP,PSTACK
012502  005737  002622                  TST     FTIME
012506  001013                          BNE     1$
012510  013737  000004  002624          MOV     @#4,SAVE4
012516  013737  000006  002626          MOV     @#6,SAVE6
012524  005037  002604                  CLR     PONE                    ; INIT FLAG

012530  012737  000001  002622          MOV     #1,FTIME
012536  013737  002624  000004  1$:     MOV     SAVE4,@#4
012544  013737  002626  000006          MOV     SAVE6,@#6
                                ;SEE IF PROGRAM JUST STARTED, BR IF YES
012552                                  READEF  #EF.START
012552  012700  000040                  MOV     #EF.START,R0
012556  104447                          TRAP    C$REFG
012560                                  BNCOMPLETE      33$
012560  103003                          BCC     33$
012562  005037  002604                  CLR     PONE                    ; RESET SWITCH
012566  000414                          BR      NEWST
012570                          33$:
                                ;SEE IF THIS IS A NEW PASS, BR IF YES
012570                                  READEF  #EF.NEW
012570  012700  000035                  MOV     #EF.NEW,R0
012574  104447                          TRAP    C$REFG
012576                                  BCOMPLETE       NEWST
012576  103410                          BCS     NEWST
                                ;SEE IF PROGRAM WAS JUST CONTINUED
012600                                  READEF  #EF.CONTINUE
012600  012700  000036                  MOV     #EF.CONTINUE,R0
012604  104447                          TRAP    C$REFG
012606                                  BCOMPLETE       ENDIT
012606  103566                          BCS     ENDIT

                                ;SEE IF PROGRAM JUST RESTARTED, BR IF NOT
012610                                  READEF  #EF.RESTART
012610  012700  000037                  MOV     #EF.RESTART,R0
012614  104447                          TRAP    C$REFG
012616                                  BNCOMPLETE      GETPRM
012616  103003                          BCC     GETPRM

012620                          NEWST:
                                ;RESET LOGICAL DEVICE TO -1
012620  012737  177777  002524          MOV     #-1,LOGDEV
```

```
                                        ;GET UNIBUS ADRS, VECTOR, PRIORITY LEVEL, LINE UNIT,SWITCH
                                        ;    PACKS, TEST CONNECTOR INFO. FOR THIS M8200,4,6,7 (CURRENT LOGICAL
                                        ;    DEVICE).
012626                                  GETPRM:
012626  005237  002524                          INC     LOGDEV
012632                                          GPHARD  LOGDEV,R1
012632  013700  002524                          MOV     LOGDEV,R0
012636  104442                                  TRAP    C$GPHRD
012640  010001                                  MOV     R0,R1
012642                                          BNCOMPLETE      GETPRM
012642  103371                                  BCC     GETPRM
                                        ;GET ADDRESS OF M8200,4,6,7
012644  012737  000006  002600                  MOV     #6,WTYPE
012652  005721                                  TST     (R1)+
012654  011137  002704                          MOV     (R1),KMCSR
                                        ;GET POINTER TO M8200,4,6,7 CSR HI BYTE
012660  011137  002706                          MOV     (R1),KMCSRH
012664  005237  002706                          INC     KMCSRH
                                        ;GET POINTER TO M8200,4,6,7 CTL OUT REG
012670  011137  002710                          MOV     (R1),KMCTL
012674  062737  000002  002710                  ADD     #2,KMCTL
                                        ;GET POINTER TO M8200,4,6,7 PORT REG - SEL 4
012702  011137  002712                          MOV     (R1),KMPO4
012706  062737  000004  002712                  ADD     #4,KMPO4
                                        ;GET POINTER TO M8200,4,6,7 PORT REG - SEL 6
012714  012137  002714                          MOV     (R1)+,KMPO6
012720  062737  000006  002714                  ADD     #6,KMPO6
                                        ;GET POINTER TO RCV VECTOR
012726  011137  002674                          MOV     (R1),KMRVEC
                                        ;GET POINTER TO RCV PRIORITY LEVEL
012732  011137  002676                          MOV     (R1),KMRLVL
012736  062737  000002  002676                  ADD     #2,KMRLVL
                                        ;GET POINTER TO TX VECTOR
012744  011137  002700                          MOV     (R1),KMTVEC
012750  062737  000004  002700                  ADD     #4,KMTVEC
                                        ;GET POINTER TO TX PRIORITY LEVEL
012756  011137  002702                          MOV     (R1),KMTLVL
012762  062737  000006  002702                  ADD     #6,KMTLVL
                                        ;PUT VECTOR INTO STAT1
012770  012137  002666                          MOV     (R1)+,STAT1
                                        ;PUT PRIORITY INTO STAT1
012774  052137  002666                          BIS     (R1)+,STAT1
                                        ;SEE IF NO LINE UNIT, SET BIT IF YES
013000  005711                                  TST     (R1)
013002  001004                                  BNE     50000$
013004  052737  010000  002666                  BIS     #BIT12,STAT1
013012  000416                                  BR      4$
013014                                          50000$:
                                        ;SEE IF M8201 LINE UNIT, SET BIT IF YES
013014  021127  000001                          CMP     (R1),#1
013020  001001                                  BNE     50001$
013022  000412                                  BR      4$
013024                                          50001$:
                                        ;SEE IF M8202 LINE UNIT, SET BIT IF YES
013024  021127  000002                          CMP     (R1),#2
013030  001004                                  BNE     50002$
```

```
2676  013032  052737  020000  002666          BIS     #BIT13,STAT1
2677  013040  000403                          BR      4$
2678  013042                                  50002$:
2679                                  ;SET BIT FOR M8203 LINE UNIT
2680  013042  052737  100000  002666          BIS     #BIT15,STAT1
2681  013050                                  4$:
2682                                  ;SET BIT IN STAT1 FOR TEST CONNECTOR
2683  013050  056137  000006  002666          BIS     6(R1),STAT1
2684  013056  062701  000002                  ADD     #2,R1
2685                                  ;SET SWITCH PACK #1 IN STAT2 LOW BYTE
2686  013062  012137  002670                  MOV     (R1)+,STAT2
2687                                  ;SET SWITCH PACK #2 IN STAT2 HIGH BYTE
2688  013066  111137  002671                  MOVB    (R1),STAT2+1
2689
2690                                  ;INCREMENT LOGICAL UNIT (DEVICE) NUMBER
2691                                  ;       INC     LOGDEV
2692  013072  000240                          NOP
2693  013074  000240                          NOP
2694
2695  013076  012737  002000  002560          MOV     #2000,MEMSZ
2696  013104  005037  002602                  CLR     TYPE
2697  013110  123727  002600  000000          CMPB    WTYPE,#0
2698  013116  001422                          BEQ     ENDIT
2699  013120  123727  002600  000004          CMPB    WTYPE,#4        ;KMC?
2700  013126  001004                          BNE     5$
2701  013130  012737  000001  002602          MOV     #1,TYPE
2702  013136  000412                          BR      ENDIT
2703  013140  012737  003777  002560  5$:     MOV     #3777,MEMSZ
2704  013146  123727  002600  000006          CMPB    WTYPE,#6
2705  013154  001003                          BNE     ENDIT
2706  013156  012737  000001  002602          MOV     #1,TYPE
2707  013164                          ENDIT:
2708  013164                                  ENDINIT
2709  013164                          L10045:
2710  013164  104411                          TRAP    C$INIT
2711
2712                                  .EVEN
2713
2714
2715
2716
2717
```

```
2718                            .SBTTL  CLEANUP CODING SECTION
2719
2720                            ;////////////////////////////////////////////////////////////////////////
2721                            ;/ THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
2722                            ;/ AT THE END OF EACH PASS.
2723                            ;////////////////////////////////////////////////////////////////////////
2724
2725  013166                            BGNCLN
2726  013166                    L$CLEAN::
2727
2728
2729  013166                            ENDCLN
2730  013166                    L10046:
2731  013166  104412                   TRAP    C$CLEAN
2732
2733
2734
2735
2736
```

```
                        .SBTTL  DROP UNIT SECTION

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

013170                          BGNDU
013170                  L$DU::
                        ;ISSUE UNIBUS RESET TO CLEAN UP
013170                          BRESET
013170  104433                  TRAP    C$RESET
013172                          ENDDU
013172                  L10047:
013172  104453                  TRAP    C$DU
```

```
2757                                    .SBTTL  ADD UNIT SECTION
2758
2759                                    ;////////////////////////////////////////////////////////////////////
2760                                    ;/ THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
2761                                    ;/  TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING.  IF
2762                                    ;/  "EF.AUNIT" IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
2763                                    ;////////////////////////////////////////////////////////////////////
2764
2765  013174                                    BGNAU
2766  013174                            L$AU::
2767  013174                                    ENDAU
2768  013174                            L10050:
2769  013174  104452                           TRAP    C$AU
2770
2771
2772
2773
2774
2775
```

```
                        .SBTTL  HARDWARE TESTS


013176                          BADHEAD
                                ;********************* TEST 1 *********************
                                ;*VERIFY THAT REFERENCING UNIBUS DEVICE REGISTERS
                                ;*DOES NOT CAUSE A TIME OUT TRAP
013176                          BADHEAD
                                ;********************* TEST 1 *********************

013176                  BGNTST
013176                  T1::
013176  013701  002704          MOV     KMCSR,R1        ;R1 CONTAINS BASE M8200,4,6,7 ADDRESS
013202  012705  000004          MOV     #4,R5           ;4 REGISTERS TO BE TESTED
013206  012737  013244  000004  MOV     #2$,4           ;SET UP TIMEOUT TRAP
013214  012737  000240  000006  MOV     #240,6          ;LEVEL 5
013222  005711          1$:     TST     (R1)            ;REFERENCE DEVICE REGISTER
013224  000240                  NOP
013226                          ESCAPE TST
013226  104410                  TRAP    C$ESCAPE
013230  000066                  .WORD   L10051-.
013232  062701  000002          ADD     #2,R1           ;NEXT REGISTER
013236  005305                  DEC     R5              ;DEC REGISTER COUNT
013240  001370                  BNE     1$              ;BR IF NOT LAST REGISTER
013242  000415                  BR      3$
013244  062706  000004  2$:     ADD     #4,SP
013250                          ERROR   1               ;TIME-OUT ERROR
013266  104455                  TRAP    C$ERDF
013270  000001                  .WORD   1
013272  004530                  .WORD   EM1
013274  006410                  .WORD   ERR1

013276  013737  002624  000004  3$:     MOV     SAVE4,4
013304  013737  002626  000006          MOV     SAVE6,6
013312                          ESCAPE  TST
013312  104410                  TRAP    C$ESCAPE
013314  000002                  .WORD   L10051-.
013316                  ENDTST
013316                  L10051:
013316  104401                  TRAP    C$ETST
```

```
013320                                  BADHEAD
                                        ;******************** TEST 2 ********************
                                        ;*VERIFY THAT RUN CAN BE CLEARED
013320                                  BADHEAD
                                        ;******************** TEST 2 ********************

013320                          BGNTST
013320                          T2::
013320                                  MYINT
013324  005011                          CLR     (R1)            ;CLEAR KMCSR
013326  005037  002612                  CLR     $GDDAT          ;CLEAR "EXPECTED"
013332  011104                          MOV     (R1),R4         ;PUT KMCSR IN "FOUND"
013334  001413                          BEQ     1$              ;BR IF CLEARED
013336                                  ERROR   26              ;ERROR KMCSR NOT CLEARED
013354  104455                          TRAP    C$ERDF
013356  000032                          .WORD   26
013360  005743                          .WORD   EM26
013362  011122                          .WORD   ERR26
013364                          1$:
013364                          ENDTST
013364                          L10052:
013364  104401                          TRAP    C$ETST
```

```
                                                BADHEAD
                                                ;******************** TEST 3 **********************
                                                ;*UNIBUS REGISTER WORD DUAL ADDRESSING TEST
                                                ;*LOAD ALL REGISTERS WITH INCREMENTING PATTERN
                                                ;*READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING
                                                ;*THE SEQUENCE:
                                                ;*      1. CLEAR REGISTER
                                                ;*      2. WRITE PATTERN
                                                ;*      3. VERIFY PATTERN
                                                ;*      4. DO ALL 4 REGISTERS
                                                ;*      5. READ ALL BACK IF ERRORS,
                                                ;*         DUAL ADDRESS PROBLEM.
                                                ;*
                                                ;*         1 IN REG 0
                                                ;*         2 IN REG 2
                                                ;*         3 IN REG 4
                                                ;*         4 IN REG 6
013366                                          BADHEAD
                                                ;******************** TEST 3 **********************

013366                                  BGNTST
013366                                  T3::
                                                MYINT
013372                                          MSTCLR                  ;R1 CONTAINS BASE M8200,4,6,7 ADDRESS
013376  012705  000001                          MOV     #1,R5           ;MASTER CLEAR M8200,4,6,7
013402                                          BGNSEG                  ;START PATTERN AT 1
013402  104404                                  TRAP    C$BSEG
013404  005011                          1$:     CLR     (R1)            ;CLEAR REGISTER
013406  010537  002612                          MOV     R5,$GDDAT               ;PUT DATA IN "EXPECTED"
013412  010511                                  MOV     R5,(R1)         ;WRITE M8200,4,6,7 REGISTER WITH PATTERN
013414  011104                                  MOV     (R1),R4         ;READ M8200,4,6,7 REGISTER INTO "FOUND"
013416  023704  002612                          CMP     $GDDAT,R4               ;IS DATA CORRECT
013422  001413                                  BEQ     2$              ;BR IS YES
013424                                          ERROR   27              ;DATA ERROR
013442  104455                                  TRAP    C$ERDF
013444  000033                                  .WORD   27
013446  005774                                  .WORD   EM27
013450  011230                                  .WORD   ERR27
013452                                  2$:     ESCAPE   SEG
013452  104410                                  TRAP    C$ESCAPE
013454  000014                                  .WORD   10000$-.
013456  005721                                  TST     (R1)+           ;NEXT REGISTER
013460  005205                                  INC     R5              ;INCREMENT DATA PATTERN
013462  022705  000005                          CMP     #5,R5           ;LAST REGISTER?
013466  001346                                  BNE     1$              ;BR IF NO
013470                                          ENDSEG
013470                                  10000$:
013470  104405                                  TRAP    C$ESEG
013472  013701  002704                          MOV     KMCSR,R1        ;BASE M8200,4,6,7 ADDRESS TO R1
013476  012705  000001                          MOV     #1,R5           ;RESTART PATTERN AT 1
013502  012737  013512  002414                  MOV     #3$,LOCK        ;NEW SCOP1
013510                                          BGNSEG
013510  104404                                  TRAP    C$BSEG
013512  010537  002612                  3$:     MOV     R5,$GDDAT               ;PUT DATA IN "EXPECTED"
013516  011104                                  MOV     (R1),R4         ;READ COMM. MICR-PROCESSOR FAMILY REGISTER INTO "FOUND"
```

```
2896  013520  023704  002612          CMP     $GDDAT,R4               ;IS DATA CORRECT
2897  013524  001413                  BEQ     4$              ;BR IF YES
2898  013526                          ERROR   2               ;DUAL ADDRESSING ERROR
2899  013544  104455                  TRAP    C$ERDF
2900  013546  000002                  .WORD   2
2901  013550  004556                  .WORD   EM2
2902  013552  006516                  .WORD   ERR2
2903  013554                  4$:     ESCAPE  SEG
2904  013554  104410                  TRAP    C$ESCAPE
2905  013556  000014                  .WORD   10001$-.
2906  013560  005721                  TST     (R1)+           ;NEXT REGISTER
2907  013562  005205                  INC     R5              ;INCREMENT PATTERN
2908  013564  022705  000005          CMP     #5,R5           ;LAST REGISTER?
2909  013570  001350                  BNE     3$              ;BR IF NO
2910  013572                          ENDSEG
2911  013572                  10001$:
2912  013572  104405                  TRAP    C$ESEG
2913  013574                  ENDTST
2914  013574                  L10053:
2915  013574  104401                  TRAP    C$ETST
```

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

                    BGNTST
                    T4::
                        MSTCLR                      ;MASTER CLEAR M8200,4,6,7
013602 005037 002576    CLR     MR0
013606 012702 000001    MOV     #BIT0,R2            ;INDICATE BSEL0
013612                  BGNSEG
013612 104404           TRAP    C$BSEG
013614 013701 002704 1$:MOV     KMCSR,R1            ;PUT REGISTER ADDRESS IN R1
013620 010237 002612    MOV     R2,$GDDAT
013624 013711 002612    MOV     $GDDAT,(R1)             ;WRITE BIT 0
013630 011104           MOV     (R1),R4             ;READ CONTROL STATUS REGISTER
013632 023704 002612    CMP     $GDDAT,R4               ;IS DATA CORRECT
013636 001413           BEQ     2$                  ;BR IF YES
013640                  ERROR   27                  ;DATA ERROR
013656 104455           TRAP    C$ERDF
013660 000033           .WORD   27
013662 005774           .WORD   EM27
013664 011230           .WORD   ERR27
013666                  ESCAPE  SEG
013666 104410           TRAP    C$ESCAPE
013670 000052           .WORD   10000$-.
013672 040211       3$: BIC     R2,(R1)             ;CLEAR BSEL0
013674 005037 002612    CLR     $GDDAT              ;CLEAR "EXPECTED"
013700 011104           MOV     (R1),R4             ;READ CONTROL STATUS REGISTER
013702 001413           BEQ     4$                  ;BR IF ZERO
013704                  ERROR   2                   ;DATA ERROR BSEL NOT CLEARED
013722 104455           TRAP    C$ERDF
013724 000002           .WORD   2
013726 004556           .WORD   EM2
013730 006516           .WORD   ERR2
013732              4$: ESCAPE  SEG
013732 104410           TRAP    C$ESCAPE
013734 000006           .WORD   10000$-.
013736 106302           ASLB    R2
013740 001325           BNE     1$
013742                  ENDSEG
013742              10000$:
013742 104405           TRAP    C$ESEG
013744              ENDTST
013744              L10054:
013744 104401           TRAP    C$ETST
```

```
013746                                          BADHEAD
                                                ;******************** TEST 5 ********************
                                                ;*CONTROL STATUS REGISTER WRITE/READ TEST
                                                ;*SET BIT9, VERIFY BIT9 WAS SET
                                                ;*CLEAR BIT9, VERIFY BIT9 WAS CLEARED
013746                                          BADHEAD
                                                ;******************** TEST 5 ********************

013746                                  BGNTST
013746                                  T5::
013746                                          MSTCLR                          ;MASTER CLEAR M8200,4,6,7
013752                                          BGNSEG
013752  104404                                  TRAP    C$BSEG
013754  013701  002704                  1$:     MOV     KMCSR,R1                ;PUT REGISTER ADDRESS IN R1
013760  012737  001000  002612                  MOV     #BIT9,$GDDAT            ;PUT DATA IN "EXPECTED"
013766  013711  002612                          MOV     $GDDAT,(R1)                     ;WRITE BIT 9
013772  011104                                  MOV     (R1),R4                 ;READ CONTROL STATUS REGISTER
013774  023704  002612                          CMP     $GDDAT,R4                       ;IS DATA CORRECT
014000  001413                                  BEQ     2$                      ;BR IF YES
014002                                          ERROR   27                      ;DATA ERROR
014020  104455                                  TRAP    C$ERDF
014022  000033                                  .WORD   27
014024  005774                                  .WORD   EM27
014026  011230                                  .WORD   ERR27
014030                                  2$:     ESCAPE  SEG
014030  104410                                  TRAP    C$ESCAPE
014032  000002                                  .WORD   10000$-.
014034                                          ENDSEG
014034                                  10000$:
014034  104405                                  TRAP    C$ESEG
014036                                          BGNSEG
014036  104404                                  TRAP    C$BSEG
014040  042711  001000                  3$:     BIC     #BIT9,(R1)              ;CLEAR BIT 9
014044  005037  002612                          CLR     $GDDAT                  ;CLEAR "EXPECTED"
014050  011104                                  MOV     (R1),R4                 ;READ CONTROL STATUS REGISTER
014052  001416                                  BEQ     4$                      ;BR IF ZERO
014054                                          ERROR   26                      ;DATA ERROR BIT9 NOT CLEARED
014072  104455                                  TRAP    C$ERDF
014074  000032                                  .WORD   26
014076  005743                                  .WORD   EM26
014100  011122                                  .WORD   ERR26
014102                                          ESCAPE  SEG
014102  104410                                  TRAP    C$ESCAPE
014104  000002                                  .WORD   10001$-.
014106                                          ENDSEG
014106                                  10001$:
014106  104405                                  TRAP    C$ESEG
014110                                  4$:
014110                                  ENDTST
014110                                  L10055:
014110  104401                                  TRAP    C$ETST
```

```
014112                                  BADHEAD
                                        ;******************* TEST 6 *******************
                                        ;*CONTROL STATUS REGISTER WRITE/READ TEST
                                        ;*SET BIT11, VERIFY BIT11 WAS SET
                                        ;*CLEAR BIT11, VERIFY BIT11 WAS CLEARED
014112                                  BADHEAD
                                        ;******************* TEST 6 *******************

014112                          BGNTST
014112                          T6::
014112                                  MSTCLR                          ;MASTER CLEAR M8200,4,6,7
014116                                  BGNSEG
014116  104404                          TRAP    C$BSEG
014120  013701  002704          1$:     MOV     KMCSR,R1                ;PUT REGISTER ADDRESS IN R1
014124  012737  004000  002612          MOV     #BIT11,$GDDAT           ;PUT DATA IN "EXPECTED"
014132  013711  002612                  MOV     $GDDAT,(R1)                     ;WRITE BIT 11
014136  011104                          MOV     (R1),R4                 ;READ CONTROL STATUS REGISTER
014140  023704  002612                  CMP     $GDDAT,R4                       ;IS DATA CORRECT
014144  001413                          BEQ     2$                      ;BR IF YES
014146                                  ERROR   26                      ;DATA ERROR
014164  104455                          TRAP    C$ERDF
014166  000032                          .WORD   26
014170  005743                          .WORD   EM26
014172  011122                          .WORD   ERR26
014174                          2$:     ESCAPE  SEG
014174  104410                          TRAP    C$ESCAPE
014176  000002                          .WORD   10000$-.
014200                                  ENDSEG
014200                          10000$:
014200  104405                          TRAP    C$ESEG
014202                                  BGNSEG
014202  104404                          TRAP    C$BSEG
014204  042711  004000          3$:     BIC     #BIT11,(R1)             ;CLEAR BIT 11
014210  005037  002612                  CLR     $GDDAT                  ;CLEAR "EXPECTED"
014214  011104                          MOV     (R1),R4                 ;READ CONTROL STATUS REGISTER
014216  001414                          BEQ     4$                      ;BR IF ZERO
014220                                  ERROR   26                      ;DATA ERROR BIT11 NOT CLEARED
014236  104455                          TRAP    C$ERDF
014240  000032                          .WORD   26
014242  005743                          .WORD   EM26
014244  011122                          .WORD   ERR26
014246                                  ENDSEG
014246                          10001$:
014246  104405                          TRAP    C$ESEG
014250                          4$:
014250                          ENDTST
014250                          L10056:
014250  104401                          TRAP    C$ETST
```

```
                                        BADHEAD
                                        ;******************** TEST 7 *********************
                                        ;*CONTROL STATUS REGISTER WRITE/READ TEST
                                        ;*SET BIT12, VERIFY BIT12 WAS SET
                                        ;*CLEAR BIT12, VERIFY BIT12 WAS CLEARED
                                        BADHEAD
                                        ;******************** TEST 7 *********************

                                BGNTST
                                T7::
                                        MSTCLR                  ;MASTER CLEAR M8200,4,6,7
                                        BGNSEG
                                        TRAP    C$BSEG
                                1$:     MOV     KMCSR,R1        ;PUT REGISTER ADDRESS IN R1
                                        MOV     #BIT12,$GDDAT   ;PUT DATA IN 'EXPECTED''
                                        MOV     $GDDAT,(R1)             ;WRITE BIT 12
                                        MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
                                        CMP     $GDDAT,R4               ;IS DATA CORRECT
                                        BEQ     2$              ;BR IF YES
                                        ERROR   26              ;DATA ERROR
                                        TRAP    C$ERDF
                                        .WORD   26
                                        .WORD   EM26
                                        .WORD   ERR26
                                2$:     ESCAPE  SEG
                                        TRAP    C$ESCAPE
                                        .WORD   10000$-.
                                        ENDSEG
                                10000$:
                                        TRAP    C$ESEG
                                        BGNSEG
                                        TRAP    C$BSEG
                                3$:     BIC     #BIT12,(R1)     ;CLEAR BIT 12
                                        CLR     $GDDAT          ;CLEAR 'EXPECTED''
                                        MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
                                        BEQ     4$              ;BR IF ZERO
                                        ERROR   26              ;DATA ERROR BIT12 NOT CLEARED
                                        TRAP    C$ERDF
                                        .WORD   26
                                        .WORD   EM26
                                        .WORD   ERR26
                                        ENDSEG
                                10001$:
                                        TRAP    C$ESEG
                                4$:
                                ENDTST
                                L10057:
                                        TRAP    C$ETST
```

```
014412                                  BADHEAD
                                        ;******************** TEST 8 *********************
                                        ;*CONTROL OUT REGISTER WRITE/READ TEST
                                        ;*FLOAT A ONE THROUGH SEL2
014412                                  BADHEAD
                                        ;******************** TEST 8 *********************

014412                          BGNTST
014412                          T8::
014412                                  MSTCLR                  ;MASTER CLEAR M8200,4,6,7
014416  012737  000002  002576          MOV     #2,MR0
014424  012702  000001                  MOV     #1,R2
014430                                  BGNSEG
014430  104404                          TRAP    C$BSEG

014432  013701  002710          1$:     MOV     KMCTL,R1        ;PUT REGISTER ADDRESS IN R1
014436  010237  002612                  MOV     R2,$GDDAT       ;PUT DATA IN "EXPECTED"
014442  013711  002612                  MOV     $GDDAT,(R1)             ;WRITE BIT 0
014446  011104                          MOV     (R1),R4         ;READ CONTROL OUT REGISTER
014450  023704  002612                  CMP     $GDDAT,R4               ;IS DATA CORRECT
014454  001413                          BEQ     2$              ;BR IF YES
014456                                  ERROR   27              ;DATA ERROR
014474  104455                          TRAP    C$ERDF
014476  000033                          .WORD   27
014500  005774                          .WORD   EM27
014502  011230                          .WORD   ERR27
014504                          2$:     ESCAPE  SEG
014504  104410                          TRAP    C$ESCAPE
014506  000052                          .WORD   10000$-.
014510  040211                  3$:     BIC     R2,(R1) ;CLEAR BIT
014512  005037  002612                  CLR     $GDDAT          ;CLEAR "EXPECTED"
014516  011104                          MOV     (R1),R4         ;READ CONTROL OUT REGISTER
014520  001413                          BEQ     4$              ;BR IF ZERO
014522                                  ERROR   27              ;DATA ERROR BIT0 NOT CLEARED
014540  104455                          TRAP    C$ERDF
014542  000033                          .WORD   27
014544  005774                          .WORD   EM27
014546  011230                          .WORD   ERR27
014550                          4$:     ESCAPEE SEG
014550  104410                          TRAP    C$ESCAPE
014552  000006                          .WORD   10000$-.
014554  006302                          ASL     R2
014556  001325                          BNE     1$
014560                                  ENDSEG
014560                          10000$:
014560  104405                          TRAP    C$ESEG
014562                          ENDTST
014562                          L10060:
014562  104401                          TRAP    C$ETST
```

```
3161  014564                                BADHEAD
3162                                        ;******************** TEST 9 ********************
3163                                        ;*PORT4 REGISTER WRITE/READ TEST
3164                                        ;*FLOAT A ONE THROUGH PORT4 REGISTER
3165                                        ;*FLOAT A ZERO THROUGH PORT4 REGISTER
3166  014564                                BADHEAD
3167                                        ;******************** TEST 9 ********************
3168
3169
3170  014564                          BGNTST
3171  014564                          T9::
3172  014564  012737  000004  002576          MOV     #4,MR0
3173  014572                                  MSTCLR                  ;MASTER CLEAR M8200,4,6,7
3174  014576  013701  002712                  MOV     KMP04,R1        ;PUT REGISTER ADDRESS IN R1
3175  014602  012705  000001                  MOV     #1,R5            ;START WITH BIT0
3176  014606                                  BGNSEG
3177  014606  104404                          TRAP    C$BSEG
3178  014610                          64$:
3179  014610  010537  002612                  MOV     R5,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
3180  014614  013711  002612                  MOV     $GDDAT,(R1)             ;WRITE PORT4 REGISTER
3181  014620  011104                          MOV     (R1),R4         ;READ PORT4 REGISTER
3182  014622  023704  002612                  CMP     $GDDAT,R4               ;COMPARE EXPECTED AND FOUND
3183  014626  001413                          BEQ     65$             ;BR IF OK
3184  014630                                  ERROR   27              ;WRITE/READ ERROR
3185  014646  104455                          TRAP    C$ERDF
3186  014650  000033                          .WORD   27
3187  014652  005774                          .WORD   EM27
3188  014654  011230                          .WORD   ERR27
3189  014656                          65$:    ESCAPE  SEG
3190  014656  104410                          TRAP    C$ESCAPE
3191  014660  000010                          .WORD   10000$-.
3192  014662  000241                          CLC                     ;CLEAR CARRY
3193  014664  006105                          ROL     R5              ;SHIFT TO NEXT BIT
3194  014666  001350                          BNE     64$             ;BR IF NOT DONE YET?
3195  014670                                  ENDSEG
3196  014670                          10000$:
3197  014670  104405                          TRAP    C$ESEG
3198  014672  012705  000001                  MOV     #1,R5           ;START WITH BIT0
3199  014676                                  BGNSEG
3200  014676  104404                          TRAP    C$BSEG
3201  014700                          66$:
3202  014700  005105                          COM     R5              ;CHANGE TO A FLOATING ZERO
3203  014702  010537  002612                  MOV     R5,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
3204  014706  013711  002612                  MOV     $GDDAT,(R1)             ;WRITE PORT4 REGISTER
3205  014712  011104                          MOV     (R1),R4
3206  014714  023704  002612                  CMP     $GDDAT,R4               ;COMPARE EXPECTED AND FOUND
3207  014720  001413                          BEQ     67$             ;BR IF OK
3208  014722                                  ERROR   27              ;WRITE/READ ERROR
3209  014740  104455                          TRAP    C$ERDF
3210  014742  000033                          .WORD   27
3211  014744  005774                          .WORD   EM27
3212  014746  011230                          .WORD   ERR27
3213  014750                          67$:    ESCAPE  SEG
3214  014750  104410                          TRAP    C$ESCAPE
3215  014752  000012                          .WORD   10001$-.
3216  014754  005105                          COM     R5              ;CHANGE BACK TO A FLOATING ONE
```

```
3217  014756  000241                    CLC                     ;CLEAR CARRY
3218  014760  006105                    ROL     R5              ;SHIFT TO NEXT BIT
3219  014762  001346                    BNE     66$             ;BR IF NOT DONE YET?
3220  014764                            ENDSEG
3221  014764                        10001$:
3222  014764  104405                    TRAP    C$ESEG
3223  014766                        ENDTST
3224  014766                        L10061:
3225  014766  104401                    TRAP    C$ETST
```

```
3226  014770                                    BADHEAD
3227                                            ;******************** TEST 10 ********************
3228                                            ;*PORT6 REGISTER WRITE/READ TEST
3229                                            ;*FLOAT A ONE THROUGH PORT6 REGISTER
3230                                            ;*FLOAT A ZERO THROUGH PORT6 REGISTER
3231  014770                                    BADHEAD
3232                                            ;******************** TEST 10 ********************
3233
3234  014770                                BGNTST
3235  014770                                T10::
3236  014770  012737  000006  002576            MOV     #6,MR0
3237  014776                                    MSTCLR                  ;MASTER CLEAR M8200,4,6,7
3238  015002  013701  002714                    MOV     KMP06,R1        ;PUT REGISTER ADDRESS IN R1
3239  015006  012705  000001                    MOV     #1,R5            ;START WITH BIT0
3240  015012                                    BGNSEG
3241  015012  104404                            TRAP    C$BSEG
3242  015014                                64$:
3243  015014  010537  002612                    MOV     R5,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
3244  015020  013711  002612                    MOV     $GDDAT,(R1)             ;WRITE PORT6 REGISTER
3245  015024  011104                            MOV     (R1),R4         ;READ PORT6 REGISTER
3246  015026  023704  002612                    CMP     $GDDAT,R4               ;COMPARE EXPECTED AND FOUND
3247  015032  001413                            BEQ     65$             ;BR IF OK
3248  015034                                    ERROR   27              ;WRITE/READ ERROR
3249  015052  104455                            TRAP    C$ERDF
3250  015054  000033                            .WORD   27
3251  015056  005774                            .WORD   EM27
3252  015060  011230                            .WORD   ERR27
3253  015062                                65$:ESCAPE  SEG
3254  015062  104410                            TRAP    C$ESCAPE
3255  015064  000010                            .WORD   10000$-.
3256  015066  000241                            CLC                     ;CLEAR CARRY
3257  015070  006105                            ROL     R5              ;SHIFT TO NEXT BIT
3258  015072  001350                            BNE     64$             ;BR IF NOT DONE YET?
3259  015074                                    ENDSEG
3260  015074                                10000$:
3261  015074  104405                            TRAP    C$ESEG
3262  015076  012705  000001                    MOV     #1,R5           ;START WITH BIT0
3263  015102                                    BGNSEG
3264  015102  104404                            TRAP    C$BSEG
3265  015104                                66$:
3266  015104  005105                            COM     R5              ;CHANGE TO A FLOATING ZERO
3267
3268  015106  010537  002612                    MOV     R5,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
3269  015112  013711  002612                    MOV     $GDDAT,(R1)             ;WRITE PORT6 REGISTER
3270  015116  011104                            MOV     (R1),R4         ;READ PORT6 REGISTER
3271  015120  023704  002612                    CMP     $GDDAT,R4               ;COMPARE EXPECTED AND FOUND
3272  015124  001413                            BEQ     67$             ;BR IF OK
3273  015126                                    ERROR   27              ;WRITE/READ ERROR
3274  015144  104455                            TRAP    C$ERDF
3275  015146  000033                            .WORD   27
3276  015150  005774                            .WORD   EM27
3277  015152  011230                            .WORD   ERR27
3278  015154                                67$:ESCAPE  SEG
3279  015154  104410                            TRAP    C$ESCAPE
3280  015156  000012                            .WORD   10001$-.
3281  015160  005105                            COM     R5              ;CHANGE BACK TO A FLOATING ONE
```

```
3282  015162  000241                    CLC                     ;CLEAR CARRY
3283  015164  006105                    ROL     R5              ;SHIFT TO NEXT BIT
3284  015166  001346                    BNE     66$             ;BR IF NOT DONE YET?
3285  015170                            ENDSEG
3286  015170                    10001$:
3287  015170  104405                    TRAP    C$ESEG
3288  015172                    ENDTST
3289  015172                    L10062:
3290  015172  104401                    TRAP    C$ETST
```

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

                                BGNTST
                                T11::
                                        MYINT
                                        MSTCLR                          ;MASTER CLEAR M8200,4,6,7
015204  012705  000001                  MOV     #1,R5                   ;START PATTERN AT 1
                                        BGNSEG
015210  104404                          TRAP    C$BSEG
015212  105011                  1$:     CLRB    (R1)                    ;CLEAR REGISTER
015214  010537  002612                  MOV     R5,$GDDAT                       ;PUT DATA IN "EXPECTED"
015220  110511                          MOVB    R5,(R1)                 ;WRITE M8200,4,6,7 REGISTER WITH PATTERN
015222  111104                          MOVB    (R1),R4                 ;READ M8200,4,6,7 REGISTER INTO "FOUND"
015224  123704  002612                  CMPB    $GDDAT,R4                       ;IS DATA CORRECT
015230  001413                          BEQ     2$                      ;BR IF YES
015232                                  ERROR   2                       ;DATA ERROR
015250  104455                          TRAP    C$ERDF
015252  000002                          .WORD   2
015254  004556                          .WORD   EM2
015256  006516                          .WORD   ERR2
015260                          2$:     ESCAPE  SEG
015260  104410                          TRAP    C$ESCAPE
015262  000024                          .WORD   10000$-.
015264  105721                          TSTB    (R1)+                   ;NEXT REGISTER
015266  005205                          INC     R5                      ;INCREMENT DATA PATTERN
015270  022705  000011                  CMP     #11,R5                  ;LAST REGISTER?
015274  001346                          BNE     1$                      ;BR IF NO
015276  013701  002704                  MOV     KMCSR,R1                ;BASE M8200,4,6,7 ADDRESS TO R1
015302  012705  000001                  MOV     #1,R5                   ;RESTART PATTERN AT 1
015306                                  ENDSEG
015306                          10000$:
015306  104405                          TRAP    C$ESEG
015310                                  BGNSEG
015310  104404                          TRAP    C$BSEG
015312  110537  002612          3$:     MOVB    R5,$GDDAT                       ;PUT DATA IN "EXPECTED"
015316  111104                          MOVB    (R1),R4                 ;READ COMM.MICRO-PROCESSOR FAMILY REGISTER INTO "FOUND"
015320  123704  002612                  CMPB    $GDDAT,R4                       ;IS DATA CORRECT
015324  001413                          BEQ     4$                      ;BR IF YES
015326                                  ERROR   2                       ;DUAL ADDRESSING ERROR
015344  104455                          TRAP    C$ERDF
015346  000002                          .WORD   2
015350  004556                          .WORD   EM2
015352  006516                          .WORD   ERR2
015354                          4$:     ESCAPE  SEG
015354  104410                          TRAP    C$ESCAPE
015356  000014                          .WORD   10001$-.
015360  105721                          TSTB    (R1)+                   ;NEXT REGISTER
015362  005205                          INC     R5                      ;INCREMENT PATTERN
015364  022705  000011                  CMP     #11,R5                  ;LAST REGISTER?
015370  001350                          BNE     3$                      ;BR IF NO
```

```
3347  015372                          ENDSEG
3348  015372                  10001$:
3349  015372  104405                  TRAP    C$ESEG
3350  015374                  ENDTST
3351  015374                  L10063:
3352  015374  104401                  TRAP    C$ETST
```

```
3353  015376                                        BADHEAD
3354                                                ;********************* TEST 12 *********************
3355                                                ;*MAINTENANCE INSTRUCTION REGISTER TEST
3356                                                ;*VERIFY THAT THE MAINT IR CAN BE WRITTEN TO ALL ZEROS'
3357                                                ;*AND ALL ONES'. VERIFY THAT IT IS CLEARED ON A BUS RESET.
3358  015376                                        BADHEAD
3359                                                ;********************* TEST 12 *********************
3360
3361  015376                                  BGNTST
3362  015376                                  T12::
3363                                                                      ;R1 CONTAINS BASE M8200,4,6,7 ADDRESS
3364  015376                                        MSTCLR                ;MASTER CLEAR M8200,4,6,7
3365  015402                                        MYINT
3366  015406                                        BGNSEG
3367  015406  104404                               TRAP    C$BSEG
3368  015410  012711  003000                       MOV     #BIT9!BIT10,(R1) ;SEL6 IS NOW THE IR
3369  015414  005037  002612                       CLR     $GDDAT          ;PUT "EXPECTED" IN $GDDAT
3370  015420  013761  002612  000006  1$:         MOV     $GDDAT,6(R1)    ;CLEAR THE IR
3371  015426  016104  000006                       MOV     6(R1),R4        ;READ THE IR
3372  015432  023704  002612                       CMP     $GDDAT,R4               ;IS IT CLEARED?
3373  015436  001413                               BEQ     2$              ;BR IF YES
3374  015440                                       ERROR   26              ;ERROR IR IS NOT CLEAR
3375  015456  104455                               TRAP    C$ERDF
3376  015460  000032                               .WORD   26
3377  015462  005743                               .WORD   EM26
3378  015464  011122                               .WORD   ERR26
3379  015466                                  2$:   ESCAPE  SEG
3380  015466  104410                               TRAP    C$ESCAPE
3381  015470  000002                               .WORD   10000$-.
3382  015472                                       ENDSEG
3383  015472                                  10000$:
3384  015472  104405                               TRAP    C$ESEG
3385  015474  012737  177777  002612               MOV     #-1,$GDDAT              ;PUT "EXPECTED" IN $GDDAT
3386  015502                                       BGNSEG
3387  015502  104404                               TRAP    C$BSEG
3388  015504  013761  002612  000006  3$:         MOV     $GDDAT,6(R1)    ;WRITE ALL ONES TO THE IR
3389  015512  016104  000006                       MOV     6(R1),R4        ;READ THE IR
3390  015516  023704  002612                       CMP     $GDDAT,R4               ;IS IT ALL ONES?
3391  015522  001413                               BEQ     4$              ;BR IF YES
3392  015524                                       ERROR   23              ;ERROR IR IS NOT = ALL ONES
3393  015542  104455                               TRAP    C$ERDF
3394  015544  000027                               .WORD   23
3395  015546  005475                               .WORD   EM23
3396  015550  010624                               .WORD   ERR23
3397  015552                                  4$:   ESCAPE  SEG
3398  015552  104410                               TRAP    C$ESCAPE
3399  015554  000002                               .WORD   10001$-.
3400  015556                                       ENDSEG
3401  015556                                  10001$:
3402  015556  104405                               TRAP    C$ESEG
3403  015560                                  ENDTST
3404  015560                                  L10064:
3405  015560  104401                               TRAP    C$ETST
```

```
3406  015562                                    BADHEAD
3407                                            ;******************** TEST 13 ********************
3408                                            ;*MICRO PROCESSOR TEST
3409                                            ;*LOAD KMP06 WITH A MICRO-PROCESSOR INSTRUCTION, CLOCK IT
3410                                            ;*VERIFY INSTRUCTION EXECUTED PROPERLY
3411                                            ;*INSTRUCTION SHOULD MOVE IBUS*4 TO IBUS*5, IBUS*4 IS ALL 1'S
3412                                            ;*AND IBUS*5 IS ALL 0'S. RESULT SHOULD BE ALL 1'S IN SEL4
3413  015562                                    BADHEAD
3414                                            ;******************** TEST 13 ********************
3415
3416  015562                              BGNTST
3417  015562                              T13::
3418  015562                                    MYINT
3419  015566                                    MSTCLR
3420  015572  012761  000377  000004           MOV     #377,4(R1)      ;PORT4 HI BYTE=1'S
3421  015600  012711  001000                    MOV     #BIT9,(R1)              ;SET ROMI
3422  015604  012761  121105  000006           MOV     #121105,6(R1)   ;INSTR TO PORT 6.
3423  015612  052711  001400                    BIS     #BIT8!BIT9,(R1) ;CLK INSTR.
3424  015616  000240                            NOP
3425  015620  012737  177777  002612           MOV     #-1,$GDDAT      ;EXPECT ALL ONES.
3426  015626  116104  000005                    MOVB    5(R1),R4        ;READ FOUND.
3427  015632  123704  002612                    CMPB    $GDDAT,R4       ;DATA CORRECT?
3428  015636  001413                            BEQ     1$
3429  015640                                    ERROR   28
3430  015656  104455                            TRAP    C$ERDF
3431  015660  000034                            .WORD   28
3432  015662  006023                            .WORD   EM28
3433  015664  011342                            .WORD   ERR28
3434
3435  015666                              1$:   ESCAPE  TST
3436  015666  104410                            TRAP    C$ESCAPE
3437  015670  000002                            .WORD   L10065-.
3438
3439  015672                              ENDTST
3440  015672                              L10065:
3441  015672  104401                            TRAP    C$ETST
```

```
3442  015674                                        BADHEAD
3443                                                ;******************** TEST 14 ********************
3444                                                ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
3445                                                ;*FLOAT A 1 THOUGH IBUS* REGISTER 0
3446                                                ;*FLOAT A 0 THROUGH IBUS* REGISTER 0
3447  015674                                        BADHEAD
3448                                                ;******************** TEST 14 ********************
3449
3450  015674                                  BGNTST
3451  015674                                  T14::
3452  015674                                        MSTCLR                  ;MASTER CLEAR M8200,4,6,7
3453  015700  012737  000000  002576                MOV     #0,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
3454  015706  012705  000001                       MOV     #1,R5           ;START WITH BIT 0
3455
3456  015712                                        MYINT
3457  015716                                        BGNSEG
3458  015716  104404                               TRAP    C$BSEG
3459  015720                                  64$:
3460  015720  010561  000004                       MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
3461  015724                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
3462  015730  121100                               121100                  ;MOV DATA TO IBUS* REGISTER 0
3463  015732                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
3464  015736  121005                               121005                  ;READ FROM IBU* REGITER 0
3465  015740  010537  002612                       MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
3466  015744  116104  000005                       MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
3467  015750  123704  002612                       CMPB    $GDDAT,R4               ;DATA CORRECT?
3468  015754  001413                               BEQ     65$             ;BR IF YES
3469  015756                                        ERROR   27              ;ERROR
3470  015774  104455                               TRAP    C$ERDF
3471  015776  000033                               .WORD   27
3472  016000  005774                               .WORD   EM27
3473  016002  011230                               .WORD   ERR27
3474  016004                                  65$:  ESCAPE  SEG
3475  016004  104410                               TRAP    C$ESCAPE
3476  016006  000010                               .WORD   10000$-.
3477  016010  000241                               CLC                     ;CLEAR CARRY
3478  016012  106105                               ROLB    R5              ;SHIFT BIT IN R5
3479  016014  001341                               BNE     64$             ;IF R5=0 THEN DONE
3480  016016                                        ENDSEG
3481  016016                                  10000$:
3482  016016  104405                               TRAP    C$ESEG
3483  016020  012705  000001                       MOV     #1,R5           ;START WITH BIT 0
3484  016024                                        BGNSEG
3485  016024  104404                               TRAP    C$BSEG
3486  016026                                  67$:
3487  016026  005105                               COM     R5
3488  016030  010561  000004                       MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
3489  016034                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
3490  016040  121100                               121100                  ;MOV DATA TO IBUS* REGISTER 0
3491  016042                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
3492  016046  121005                               121005                  ;READ FROM IBUS* REGISTER 0
3493  016050  010537  002612                       MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
3494  016054  116104  000005                       MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
3495  016060  123704  002612                       CMPB    $GDDAT,R4               ;DATA CORRECT?
3496  016064  001413                               BEQ     68$             ;BR IF YES
3497  016066                                        ERROR   27              ;ERROR
```

```
3498   016104   104455                  TRAP    C$ERDF
3499   016106   000033                  .WORD   27
3500   016110   005774                  .WORD   EM27
3501   016112   011230                  .WORD   ERR27
3502   016114                   68$:    ESCAPE  SEG
3503   016114   104410                  TRAP    C$ESCAPE
3504   016116   000012                  .WORD   10001$-.
3505   016120   005105                  COM     R5              ;CHANGE TO FLOATING 1
3506   016122   000241                  CLC                     ;CLEAR CARRY
3507   016124   106105                  ROLB    R5              ;SHIFT BIT IN R5
3508   016126   001337                  BNE     67$             ;IF R5=0 THEN DONE
3509   016130                           ENDSEG
3510   016130                   10001$:
3511   016130   104405                  TRAP    C$ESEG
3512   016132                   ENDTST
3513   016132                   L10066:
3514   016132   104401                  TRAP    C$ETST
```

```
016134                                  BADHEAD
                                        ;******************** TEST 15 **********************
                                        ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
                                        ;*FLOAT A 1 THOUGH IBUS* REGISTER 1
                                        ;*FLOAT A 0 THROUGH IBUS* REGISTER 1
016134                                  BADHEAD
                                        ;******************** TEST 15 **********************

016134                          BGNTST
016134                          T15::
016134                                  MSTCLR                  ;MASTER CLEAR M8200,4,6,7
016140  012737  000001  002576          MOV     #1,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
016146  012705  000001                  MOV     #1,R5           ;START WITH BIT 0

016152                                  MYINT
016156                                  BGNSEG
016156  104404                          TRAP    C$BSEG
016160                          64$:
016160  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
016164  042761  000307  000004          BIC     #307,4(R1)      ; MASK
016172                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
016176  121101                          121100!1                ;MOV DATA TO IBUS* REGISTER 1
016200                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
016204  121025                          121005!<1*20>           ;READ FROM IBUS* REGISTER 1
016206  010537  002612                  MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
016212  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
016216  042737  000307  002612          BIC     #307,$GDDAT
016224  042704  000307                  BIC     #307,R4
016230  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
016234  001413                          BEQ     65$             ;BR IF YES
016236                                  ERROR   27              ;ERROR
016254  104455                          TRAP    C$ERDF
016256  000033                          .WORD   27
016260  005774                          .WORD   EM27
016262  011230                          .WORD   ERR27
016264  000241                  65$:    CLC                     ;CLEAR CARRY
016266  106105                          ROLB    R5              ;SHIFT BIT IN R5
016270  001333                          BNE     64$             ;IF R5=0 THEN DONE
016272                                  ENDSEG
016272                          10000$:
016272  104405                          TRAP    C$ESEG
016274  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
016300                                  BGNSEG
016300  104404                          TRAP    C$BSEG
016302                          67$:
016302  005105                          COM     R5
016304  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
016310  042761  000307  000004          BIC     #307,4(R1)      ; MASK
016316                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
016322  121101                          121100!1                ;MOV DATA TO IBUS* REGISTER 1
016324                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
016330  121025                          121005!<1*20>           ;READ FROM IBUS* REGISTER 1
016332  010537  002612                  MOV     R5,$GDDAT       ;PUT EXPECTED IN $GDDAT
016336  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
016342  042737  000307  002612          BIC     #307,$GDDAT
016350  042704  000307                  BIC     #307,R4
```

```
3571  016354  123704  002612          CMPB    $GDDAT,R4               ;DATA CORRECT?
3572  016360  001413                  BEQ     68$             ;BR IF YES
3573  016362                          ERROR   27              ;ERROR
3574  016400  104455                  TRAP    C$ERDF
3575  016402  000033                  .WORD   27
3576  016404  005774                  .WORD   EM27
3577  016406  011230                  .WORD   ERR27
3578  016410                  68$:    ESCAPE  SEG
3579  016410  104410                  TRAP    C$ESCAPE
3580  016412  000012                  .WORD   10001$-.
3581  016414  005105                  COM     R5              ;CHANGE TO FLOATING 1
3582  016416  000241                  CLC                     ;CLEAR CARRY
3583  016420  106105                  ROLB    R5              ;SHIFT BIT IN R5
3584  016422  001327                  BNE     67$             ;IF R5=0 THEN DONE
3585  016424                          ENDSEG
3586  016424                  10001$:
3587  016424  104405                  TRAP    C$ESEG
3588  016426                  ENDTST
3589  016426                  L10067:
3590  016426  104401                  TRAP    C$ETST
```

```
016430                                    BADHEAD
                                          ;******************** TEST 16 ********************
                                          ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
                                          ;*FLOAT A 1 THOUGH IBUS* REGISTER 2
                                          ;*FLOAT A 0 THROUGH IBUS* REGISTER 2
016430                                    BADHEAD
                                          ;******************** TEST 16 ********************

016430                           BGNTST
016430                           T16::
016430                                    MSTCLR                      ;MASTER CLEAR M8200,4,6,7
016434  012737  000002  002576            MOV     #2,MR0              ;SAVE REGISTER ADDRESS FOR TYPEOUT
016442  012705  000001                    MOV     #1,R5               ;START WITH BIT 0
016446                                    MYINT
016452                                    BGNSEG
016452  104404                            TRAP    C$BSEG
016454                           64$:
016454  010561  000004                    MOV     R5,4(R1)            ;PUT PATTERN INTO PORT4
016460                                    ROMCLK                      ;NEXT WORD IS INSTRUCTION,  BBN
016464  121102                            121100!2                    ;MOV DATA TO IBUS* REGISTER 0
016466                                    ROMCLK                      ;NEXT WORD IS INSTRUCTION,  BBN
016472  121045                            121005!<2*20>               ;READ FROM IBUS* REGISTER 2
016474  010537  002612                    MOV     R5,$GDDAT                   ;PUT EXPECTED IN $GDDAT
016500  116104  000005                    MOVB    5(R1),R4            ;PUT "FOUND" INTO R4
016504  123704  002612                    CMPB    $GDDAT,R4                   ;DATA CORRECT?
016510  001413                            BEQ     65$                 ;BR IF YES
016512                                    ERROR   27                  ;ERROR
016530  104455                            TRAP    C$ERDF
016532  000033                            .WORD   27
016534  005774                            .WORD   EM27
016536  011230                            .WORD   ERR27
016540                           65$:     ESCAPE  SEG
016540  104410                            TRAP    C$ESCAPE
016542  000010                            .WORD   10000$-.
016544  000241                            CLC                         ;CLEAR CARRY
016546  106105                            ROLB    R5                  ;SHIFT BIT IN R5
016550  001341                            BNE     64$                 ;IF R5=0 THEN DONE
016552                                    ENDSEG
016552                           10000$:
016552  104405                            TRAP    C$ESEG
016554  012705  000001                    MOV     #1,R5               ;START WITH BIT 0
016560                                    BGNSEG
016560  104404                            TRAP    C$BSEG
016562                           67$:
016562  005105                            COM     R5
016564  010561  000004                    MOV     R5,4(R1)            ;PUT PATTERN INTO PORT4
016570                                    ROMCLK                      ;NEXT WORD IS INSTRUCTION,  BBN
016574  121102                            121100!2                    ;MOV DATA TO IBUS* REGISTER 2
016576                                    ROMCLK                      ;NEXT WORD IS INSTRUCTION,  BBN
016602  121045                            121005!<2*20>               ;READ FROM IBUS* REGISTER 2
016604  010537  002612                    MOV     R5,$GDDAT                   ;PUT EXPECTED IN $GDDAT
016610  116104  000005                    MOVB    5(R1),R4            ;PUT "FOUND" INTO R4
016614  123704  002612                    CMPB    $GDDAT,R4                   ;DATA CORRECT?
016620  001413                            BEQ     68$                 ;BR IF YES
016622                                    ERROR   27                  ;ERROR
016640  104455                            TRAP    C$ERDF
```

```
3647  016642  000033                    .WORD   27
3648  016644  005774                    .WORD   EM27
3649  016646  011230                    .WORD   ERR27
3650  016650                    68$:    ESCAPE  SEG
3651  016650  104410                    TRAP    C$ESCAPE
3652  016652  000012                    .WORD   10001$-.
3653  016654  005105                    COM     R5              ;CHANGE TO FLOATING 1
3654  016656  000241                    CLC                     ;CLEAR CARRY
3655  016660  106105                    ROLB    R5              ;SHIFT BIT IN R5
3656  016662  001337                    BNE     67$             ;IF R5=0 THEN DONE
3657  016664                            ENDSEG
3658  016664                    10001$:
3659  016664  104405                    TRAP    C$ESEG
3660  016666                    ENDTST
3661  016666                    L10070:
3662  016666  104401                    TRAP    C$ETST
```

```
016670                                  BADHEAD
                                        ;******************** TEST 17 ********************
                                        ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
                                        ;*FLOAT A 1 THOUGH IBUS* REGISTER 3
                                        ;*FLOAT A 0 THROUGH IBUS* REGISTER 3
016670                                  BADHEAD
                                        ;******************** TEST 17 ********************

016670                          BGNTST
016670                          T17::
016670                                  MSTCLR                  ;MASTER CLEAR M8200,4,6,7
016674  012737  000003  002576          MOV     #3,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
016702  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
016706                                  MYINT
016712                                  BGNSEG
016712  104404                          TRAP    C$BSEG
016714                          64$:
016714  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
016720                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
016724  121103                          121100!3                ;MOV DATA TO IBUS* REGISTER 3
016726                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
016732  121065                          121005!<3*20>           ;READ FROM IBUS* REGISTER 3
016734  010537  002612                  MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
016740  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
016744  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
016750  001413                          BEQ     65$             ;BR IF YES
016752                                  ERROR   27              ;ERROR
016770  104455                          TRAP    C$ERDF
016772  000033                          .WORD   27
016774  005774                          .WORD   EM27
016776  011230                          .WORD   ERR27
017000                          65$:    ESCAPE  SEG
017000  104410                          TRAP    C$ESCAPE
017002  000010                          .WORD   10000$-.
017004  000241                          CLC                     ;CLEAR CARRY
017006  106105                          ROLB    R5              ;SHIFT BIT IN R5
017010  001341                          BNE     64$             ;IF R5=0 THEN DONE
017012                                  ENDSEG
017012                          10000$:
017012  104405                          TRAP    C$ESEG
017014  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
017020                                  BGNSEG
017020  104404                          TRAP    C$BSEG
017022                          67$:
017022  005105                          COM     R5
017024  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
017030                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
017034  121103                          121100!3                ;MOV DATA TO IBUS* REGISTER 3
017036                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
017042  121065                          121005!<3*20>           ;READ FROM IBUS* REGISTER 3
017044  010537  002612                  MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
017050  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
017054  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
017060  001413                          BEQ     68$             ;BR IF YES
017062                                  ERROR   27              ;ERROR
017100  104455                          TRAP    C$ERDF
```

```
3719  017102  000033                  .WORD   27
3720  017104  005774                  .WORD   EM27
3721  017106  011230                  .WORD   ERR27
3722  017110                  68$:    ESCAPE  SEG
3723  017110  104410                  TRAP    C$ESCAPE
3724  017112  000012                  .WORD   10001$-.
3725  017114  005105                  COM     R5              ;CHANGE TO FLOATING 1
3726  017116  000241                  CLC                     ;CLEAR CARRY
3727  017120  106105                  ROLB    R5              ;SHIFT BIT IN R5
3728  017122  001337                  BNE     67$             ;IF R5=0 THEN DONE
3729  017124                          ENDSEG
3730  017124                  10001$:
3731  017124  104405                  TRAP    C$ESEG
3732  017126                  ENDTST
3733  017126                  L10071:
3734  017126  104401                  TRAP    C$ETST
```

```
017130                                  BADHEAD
                                        ;******************** TEST 18 **********************
                                        ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
                                        ;*FLOAT A 1 THOUGH IBUS* REGISTER 4
                                        ;*FLOAT A 0 THROUGH IBUS* REGISTER 4
017130                                  BADHEAD
                                        ;******************** TEST 18 **********************

017130                          BGNTST
017130                          T18::
                                        MSTCLR                          ;MASTER CLEAR M8200,4,6,7
017134  012737  000004  002576          MOV     #4,MR0                  ;SAVE REGISTER ADDRESS FOR TYPEOUT
017142  012705  000001                  MOV     #1,R5                   ;START WITH BIT 0
017146                                  MYINT
017152                                  BGNSEG
017152  104404                          TRAP    C$BSEG
017154                          64$:
017154  010561  000004                  MOV     R5,4(R1)                ;PUT PATTERN INTO PORT4
017160                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
017164  121104                          121100!4                        ;MOV DATA TO IBUS* REGISTER 4
017166                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
017172  121105                          121005!<4*20>                   ;READ FROM IBUS* REGISTER 4
017174  010537  002612                  MOV     R5,$GDDAT                       ;PUT EXPECTED IN $GDDAT
017200  116104  000005                  MOVB    5(R1),R4                ;PUT "FOUND" INTO R4
017204  123704  002612                  CMPB    $GDDAT,R4                       ;DATA CORRECT?
017210  001413                          BEQ     65$                     ;BR IF YES
017212                                  ERROR   27                      ;ERROR
017230  104455                          TRAP    C$ERDF
017232  000033                          .WORD   27
017234  005774                          .WORD   EM27
017236  011230                          .WORD   ERR27
017240                          65$:    ESCAPE  SEG
017240  104410                          TRAP    C$ESCAPE
017242  000010                          .WORD   10000$-.
017244  000241                          CLC                             ;CLEAR CARRY
017246  106105                          ROLB    R5                      ;SHIFT BIT IN R5
017250  001341                          BNE     64$                     ;IF R5=0 THEN DONE
017252                                  ENDSEG
017252                          10000$:
017252  104405                          TRAP    C$ESEG
017254  012705  000001                  MOV     #1,R5                   ;START WITH BIT 0
017260                                  BGNSEG
017260  104404                          TRAP    C$BSEG
017262                          67$:
017262  005105                          COM     R5
017264  010561  000004                  MOV     R5,4(R1)                ;PUT PATTERN INTO PORT4
017270                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
017274  121104                          121100!4                        ;MOV DATA TO IBUS* REGISTER 4
017276                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
017302  121105                          121005!<4*20>                   ;READ FROM IBUS* REGISTER 4
017304  010537  002612                  MOV     R5,$GDDAT                       ;PUT EXPECTED IN $GDDAT
017310  116104  000005                  MOVB    5(R1),R4                ;PUT "FOUND" INTO R4
017314  123704  002612                  CMPB    $GDDAT,R4                       ;DATA CORRECT?
017320  001413                          BEQ     68$                     ;BR IF YES
017322                                  ERROR   27                      ;ERROR
017340  104455                          TRAP    C$ERDF
```

```
3791  017342  000033                    .WORD   27
3792  017344  005774                    .WORD   EM27
3793  017346  011230                    .WORD   ERR27
3794  017350                    68$:    ESCAPE  SEG
3795  017350  104410                    TRAP    C$ESCAPE
3796  017352  000012                    .WORD   10001$-.
3797  017354  005105                    COM     R5              ;CHANGE TO FLOATING 1
3798  017356  000241                    CLC                     ;CLEAR CARRY
3799  017360  106105                    ROLB    R5              ;SHIFT BIT IN R5
3800  017362  001337                    BNE     67$             ;IF R5=0 THEN DONE
3801  017364                            ENDSEG
3802  017364                    10001$:
3803  017364  104405                    TRAP    C$ESEG
3804  017366                    ENDTST
3805  017366                    L10072:
3806  017366  104401                    TRAP    C$ETST
```

```
017370                                  BADHEAD
                                        ;******************* TEST 19 *******************
                                        ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
                                        ;*FLOAT A 1 THOUGH IBUS* REGISTER 5
                                        ;*FLOAT A 0 THROUGH IBUS* REGISTER 5
017370                                  BADHEAD
                                        ;******************* TEST 19 *******************

017370                          BGNTST
017370                          T19::
017370                                  MSTCLR                          ;MASTER CLEAR M8200,4,6,7
017374  012737  000005  002576          MOV     #5,MR0                  ;SAVE REGISTER ADDRESS FOR TYPEOUT
017402  012705  000001                  MOV     #1,R5                   ;START WITH BIT 0
017406                                  MYINT
017412                                  BGNSEG
017412  104404                          TRAP    C$BSEG
017414                          64$:
017414  010561  000004                  MOV     R5,4(R1)                ;PUT PATTERN INTO PORT4
017420                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
017424  121105                          121100!5                        ;MOV DATA TO IBUS* REGISTER 5
017426                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
017432  121125                          121005!<5*20>                   ;READ FROM IBUS* REGISTER 5
017434  010537  002612                  MOV     R5,$GDDAT                       ;PUT EXPECTED IN $GDDAT
017440  116104  000005                  MOVB    5(R1),R4                ;PUT "FOUND" INTO R4
017444  123704  002612                  CMPB    $GDDAT,R4                       ;DATA CORRECT?
017450  001413                          BEQ     65$                     ;BR IF YES
017452                                  ERROR   27                      ;ERROR
017470  104455                          TRAP    C$ERDF
017472  000033                          .WORD   27
017474  005774                          .WORD   EM27
017476  011230                          .WORD   ERR27
017500                          65$:    ESCAPE  SEG
017500  104410                          TRAP    C$ESCAPE
017502  000010                          .WORD   10000$-.
017504  000241                          CLC                             ;CLEAR CARRY
017506  106105                          ROLB    R5                      ;SHIFT BIT IN R5
017510  001341                          BNE     64$                     ;IF R5=0 THEN DONE
017512                                  ENDSEG
017512                          10000$:
017512  104405                          TRAP    C$ESEG
017514  012705  000001                  MOV     #1,R5                   ;START WITH BIT 0
017520                                  BGNSEG
017520  104404                          TRAP    C$BSEG
017522                          67$:
017522  005105                          COM     R5
017524  010561  000004                  MOV     R5,4(R1)                ;PUT PATTERN INTO PORT4
017530                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
017534  121105                          121100!5                        ;MOV DATA TO IBUS* REGISTER 5
017536                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
017542  121125                          121005!<5*20>                   ;READ FROM IBUS* REGISTER 5
017544  010537  002612                  MOV     R5,$GDDAT                       ;PUT EXPECTED IN $GDDAT
017550  116104  000005                  MOVB    5(R1),R4                ;PUT "FOUND" INTO R4
017554  123704  002612                  CMPB    $GDDAT,R4                       ;DATA CORRECT?
017560  001413                          BEQ     68$                     ;BR IF YES
017562                                  ERROR   27                      ;ERROR
017600  104455                          TRAP    C$ERDF
```

```
3863  017602  000033                  .WORD   27
3864  017604  005774                  .WORD   EM27
3865  017606  011230                  .WORD   ERR27
3866  017610                  68$:    ESCAPE  SEG
3867  017610  104410                  TRAP    C$ESCAPE
3868  017612  000012                  .WORD   10001$-.
3869  017614  005105                  COM     R5              ;CHANGE TO FLOATING 1
3870  017616  000241                  CLC                     ;CLEAR CARRY
3871  017620  106105                  ROLB    R5              ;SHIFT BIT IN R5
3872  017622  001337                  BNE     67$             ;IF R5=0 THEN DONE
3873  017624                          ENDSEG
3874  017624                  10001$:
3875  017624  104405                  TRAP    C$ESEG
3876  017626                  ENDTST
3877  017626                  L10073:
3878  017626  104401                  TRAP    C$ETST
```

```
3879  017630                                    BADHEAD
3880                                            ;******************** TEST 20 ********************
3881                                            ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
3882                                            ;*FLOAT A 1 THOUGH IBUS* REGISTER 10
3883                                            ;*FLOAT A 0 THROUGH IBUS* REGISTER 10
3884  017630                                    BADHEAD
3885                                            ;******************** TEST 20 ********************
3886
3887  017630                                BGNTST
3888  017630                                T20::
3889  017630                                        MSTCLR                  ;MASTER CLEAR M8200,4,6,7
3890  017634  012737  000010  002576                MOV     #10,MR0         ;SAVE REGISTER ADDRESS FOR TYPEOUT
3891  017642  012705  000001                        MOV     #1,R5           ;START WITH BIT 0
3892  017646                                        MYINT
3893  017652                                        BGNSEG
3894  017652  104404                               TRAP    C$BSEG
3895  017654                                64$:
3896  017654  010561  000004                        MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
3897  017660  042761  000141  000004               BIC     #141,4(R1)      ;CLEAR UNWANTED BITS
3898  017666                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
3899  017672  121110                               121100!10               ;MOV DATA TO IBUS* REGISTER 10
3900  017674                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
3901  017700  121205                               121005!<10*20>          ;READ FROM IBUS* REGISTER 10
3902  017702  010537  002612                        MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
3903  017706  042737  000141  002612               BIC     #141,$GDDAT             ;CLEAR UNWANTED BITS
3904  017714  116104  000005                        MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
3905  017720  123704  002612                        CMPB    $GDDAT,R4               ;DATA CORRECT?
3906  017724  001415                               BEQ     65$             ;BR IF YES
3907  017726                                        ERROR   27              ;ERROR
3908  017744  104455                               TRAP    C$ERDF
3909  017746  000033                               .WORD   27
3910  017750  005774                               .WORD   EM27
3911  017752  011230                               .WORD   ERR27
3912  017754                                        ESCAPE  SEG
3913  017754  104410                               TRAP    C$ESCAPE
3914  017756  000010                               .WORD   10000$-.
3915  017760  000241                        65$:    CLC                     ;CLEAR CARRY
3916  017762  106105                               ROLB    R5              ;SHIFT BIT IN R5
3917  017764  001333                               BNE     64$             ;IF R5=0 THEN DONE
3918  017766                                        ENDSEG
3919  017766                                10000$:
3920  017766  104405                               TRAP    C$ESEG
3921  017770  012705  000001                        MOV     #1,R5           ;START WITH BIT 0
3922  017774                                        BGNSEG
3923  017774  104404                               TRAP    C$BSEG
3924  017776                                67$:
3925  017776  005105                               COM     R5
3926  020000  010561  000004                        MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
3927  020004  042761  000141  000004               BIC     #141,4(R1)      ;CLEAR UNWANTED BITS
3928  020012                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
3929  020016  121110                               121100!10               ;MOV DATA TO IBUS* REGISTER 10
3930  020020                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
3931  020024  121205                               121005!<10*20>          ;READ FROM IBUS* REGISTER 10
3932  020026  010537  002612                        MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
3933  020032  042737  000141  002612               BIC     #141,$GDDAT             ;CLEAR UNWANTED BITS
3934  020040  116104  000005                        MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
```

```
 3935  020044  123704  002612          CMPB    $GDDAT,R4               ;DATA CORRECT?
 3936  020050  001415                  BEQ     68$             ;BR IF YES
 3937  020052                          ERROR   27              ;ERROR
 3938  020070  104455                  TRAP    C$ERDF
 3939  020072  000033                  .WORD   27
 3940  020074  005774                  .WORD   EM27
 3941  020076  011230                  .WORD   ERR27
 3942  020100                          ESCAPE  SEG
 3943  020100  104410                  TRAP    C$ESCAPE
 3944  020102  000012                  .WORD   10001$-.
 3945  020104  005105          68$:    COM     R5              ;CHANGE TO FLOATING 1
 3946  020106  000241                  CLC                     ;CLEAR CARRY
 3947  020110  106105                  ROLB    R5              ;SHIFT BIT IN R5
 3948  020112  001331                  BNE     67$             ;IF R5=0 THEN DONE
 3949  020114                          ENDSEG
 3950  020114                  10001$:
 3951  020114  104405                  TRAP    C$ESEG
 3952  020116                  ENDTST
 3953  020116                  L10074:
 3954  020116  104401                  TRAP    C$ETST
```

```
020120                                      BADHEAD
                                            ;******************** TEST 21 ********************
                                            ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
                                            ;*FLOAT A 1 THOUGH IBUS* REGISTER 11
                                            ;*FLOAT A 0 THROUGH IBUS* REGISTER 11
020120                                      BADHEAD
                                            ;******************** TEST 21 ********************

020120                              BGNTST
020120                              T21::
020120                                      MSTCLR                          ;MASTER CLEAR M8200.4.6.7
020124  012737  000011  002576              MOV     #11,MR0                 ;SAVE REGISTER ADDRESS FOR TYPEOUT
020132  012705  000001                      MOV     #1,R5                   ;START WITH BIT 0
020136                                      MYINT
020142                                      BGNSEG
020142  104404                              TRAP    C$BSEG
020144                              64$:
020144  010561  000004                      MOV     R5,4(R1)                ;PUT PATTERN INTO PORT4
020150  042761  000222  000004              BIC     #222,4(R1)              ;CLEAR UNWANTED BITS
020156                                      ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
020162  121111                              121100!11                       ;MOV DATA TO IBUS* REGISTER 11
020164                                      ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
020170  121225                              121005!<11*20>                  ;READ FROM IBUS* REGISTER 11
020172  010537  002612                      MOV     R5,$GDDAT                       ;PUT EXPECTED IN $GDDAT
020176  042737  000222  002612              BIC     #222,$GDDAT                     ;CLEAR UNWANTED BITS
020204  116104  000005                      MOVB    5(R1),R4                ;PUT ''FOUND'' INTO R4
020210  042704  000020                      BIC     #20,R4
020214  123704  002612                      CMPB    $GDDAT,R4                       ;DATA CORRECT?
020220  001415                              BEQ     65$                     ;BR IF YES
020222                                      ERROR   27                      ;ERROR
020240  104455                              TRAP    C$ERDF
020242  000033                              .WORD   27
020244  005774                              .WORD   EM27
020246  011230                              .WORD   ERR27
020250                                      ESCAPE  SEG
020250  104410                              TRAP    C$ESCAPE
020252  000010                              .WORD   10000$-.
020254  000241                      65$:    CLC                             ;CLEAR CARRY
020256  106105                              ROLB    R5                      ;SHIFT BIT IN R5
020260  001331                              BNE     64$                     ;IF R5=0 THEN DONE
020262                                      ENDSEG
020262                              10000$:
020262  104405                              TRAP    C$ESEG
020264  012705  000001                      MOV     #1,R5                   ;START WITH BIT 0
020270                                      BGNSEG
020270  104404                              TRAP    C$BSEG
020272                              67$:
020272  005105                              COM     R5
020274  010561  000004                      MOV     R5,4(R1)                ;PUT PATTERN INTO PORT4
020300  042761  000222  000004              BIC     #222,4(R1)              ;CLEAR UNWANTED BITS
020306                                      ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
020312  121111                              121100!11                       ;MOV DATA TO IBUS* REGISTER 11
020314                                      ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
020320  121225                              121005!<11*20>                  ;READ FROM IBUS* REGISTER 11
020322  010537  002612                      MOV     R5,$GDDAT                       ;PUT EXPECTED IN $GDDAT
020326  042737  000222  002612              BIC     #222,$GDDAT                     ;CLEAR UNWANTED BITS
```

```
4011  020334  052737  000020  002612          BIS     #20,$GDDAT              ;ADD THESE BITS
4012  020342  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
4013  020346  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
4014  020352  001415                          BEQ     68$             ;BR IF YES
4015  020354                                  ERROR   27              ;ERROR
4016  020372  104455                          TRAP    C$ERDF
4017  020374  000033                          .WORD   27
4018  020376  005774                          .WORD   EM27
4019  020400  011230                          .WORD   ERR27
4020  020402                                  ESCAPE  SEG
4021  020402  104410                          TRAP    C$ESCAPE
4022  020404  000012                          .WORD   10001$-.
4023  020406  005105                  68$:    COM     R5                      ;CHANGE TO FLOATING 1
4024  020410  000241                          CLC                             ;CLEAR CARRY
4025  020412  106105                          ROLB    R5                      ;SHIFT BIT IN R5
4026  020414  001326                          BNE     67$                     ;IF R5=0 THEN DONE
4027  020416                                  ENDSEG
4028  020416                          10001$:
4029  020416  104405                          TRAP    C$ESEG
4030  020420                          ENDTST
4031  020420                          L10075:
4032  020420  104401                          TRAP    C$ETST
```

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

                                BGNTST
                                T22::
                                        MSTCLR                          ;MASTER CLEAR COMM. MICRO-PROCCESSOR FAMILY
020426  012737  000000  002576          MOV     #0,MR0                  ;SAVE REGISTER ADDRESS FOR TYPEOUT
020434  012705  000001                  MOV     #1,R5                   ;START WITH BIT 0
                                        MYINT
                                        BGNSEG
020444  104404                          TRAP    C$BSEG
                                64$:
020446  010561  000004                  MOV     R5,4(R1)                ;PUT PATTERN INTO PORT4
                                        ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
020456  122100                          122100                          ;MOV DATA TO IBUS* REGISTER 0
                                        ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
020464  021005                          21005                           ;READ FROM IBUS* REGISTER 0
020466  010537  002612                  MOV     R5,$GDDAT                       ;PUT EXPECTED IN $GDDAT
020472  116104  000005                  MOVB    5(R1),R4                ;PUT "FOUND" INTO R4
020476  123704  002612                  CMPB    $GDDAT,R4                       ;DATA CORRECT?
020502  001413                          BEQ     65$                     ;BR IF YES
020504                                  ERROR   29                      ;ERROR
020522  104455                          TRAP    C$ERDF
020524  000035                          .WORD   29
020526  006054                          .WORD   EM29
020530  011450                          .WORD   ERR29
020532                          65$:    ESCAPE  SEG
020532  104410                          TRAP    C$ESCAPE
020534  000010                          .WORD   10000$-.
020536  000241                          CLC                             ;CLEAR CARRY
020540  106105                          ROLB    R5                      ;SHIFT BIT IN R5
020542  001341                          BNE     64$                     ;IF R5=0 THEN DONE
020544                                  ENDSEG
020544                          10000$:
020544  104405                          TRAP    C$ESEG
020546  012705  000001                  MOV     #1,R5                   ;START WITH BIT 0
020552                                  BGNSEG
020552  104404                          TRAP    C$BSEG
020554                          67$:
020554  005105                          COM     R5
020556  010561  000004                  MOV     R5,4(R1)                ;PUT PATTERN INTO PORT4
020562                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
020566  122100                          122100                          ;MOV DATA TO IBUS* REGISTER 0
020570                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
020574  021005                          21005                           ;READ FROM IBUS* REGISTER 0
020576  010537  002612                  MOV     R5,$GDDAT                       ;PUT EXPECTED IN $GDDAT
020602  116104  000005                  MOVB    5(R1),R4                ;PUT "FOUND" INTO R4
020606  123704  002612                  CMPB    $GDDAT,R4                       ;DATA CORRECT?
020612  001413                          BEQ     68$                     ;BR IF YES
020614                                  ERROR   29                      ;ERROR
020632  104455                          TRAP    C$ERDF
```

```
4089  020634  000035                  .WORD   29
4090  020636  006054                  .WORD   EM29
4091  020640  011450                  .WORD   ERR29
4092  020642                  68$:    ESCAPE  SEG
4093  020642  104410                  TRAP    C$ESCAPE
4094  020644  000012                  .WORD   10001$-.
4095  020646  005105                  COM     R5              ;CHANGE TO FLOATING 1
4096  020650  000241                  CLC                     ;CLEAR CARRY
4097  020652  106105                  ROLB    R5              ;SHIFT BIT IN R5
4098  020654  001337                  BNE     67$             ;IF R5=0 THEN DONE
4099  020656                          ENDSEG
4100  020656                  10001$:
4101  020656  104405                  TRAP    C$ESEG
4102  020660                  ENDTST
4103  020660                  L10076:
4104  020660  104401                  TRAP    C$ETST
```

```
 4105  020662                                    BADHEAD
 4106                                            ;********************* TEST 23 *********************
 4107                                            ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
 4108                                            ;*FLOAT A 1 THOUGH IBUS REGISTER 1
 4109                                            ;*FLOAT A 0 THROUGH IBUS REGISTER 1
 4110  020662                                    BADHEAD
 4111                                            ;********************* TEST 23 *********************
 4112
 4113  020662                                BGNTST
 4114  020662                                T23::
 4115  020662                                        MSTCLR
 4116  020666  012737  000001  002576                MOV     #1,MR0          ;MASTER CLEAR COMM. MICRO-PROCCESSOR FAMILY
 4117  020674  012705  000001                        MOV     #1,R5           ;SAVE REGISTER ADDRESS FOR TYPEOUT
 4118  020700                                        MYINT                   ;START WITH BIT 0
 4119  020704                                        BGNSEG
 4120  020704  104404                               TRAP    C$BSEG
 4121  020706                                64$:
 4122  020706  010561  000004                        MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4123  020712                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4124  020716  122101                               122100!1                ;MOV DATA TO IBUS* REGISTER 1
 4125  020720                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4126  020724  021025                               21005!<1*20>            ;READ FROM IBUS* REGISTER 1
 4127  020726  010537  002612                        MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
 4128  020732  116104  000005                        MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 4129  020736  123704  002612                        CMPB    $GDDAT,R4               ;DATA CORRECT?
 4130  020742  001413                               BEQ     65$             ;BR IF YES
 4131  020744                                        ERROR   29              ;ERROR
 4132  020762  104455                               TRAP    C$ERDF
 4133  020764  000035                               .WORD   29
 4134  020766  006054                               .WORD   EM29
 4135  020770  011450                               .WORD   ERR29
 4136  020772                                65$:    ESCAPE  SEG
 4137  020772  104410                               TRAP    C$ESCAPE
 4138  020774  000010                               .WORD   10000$-.
 4139  020776  000241                               CLC                     ;CLEAR CARRY
 4140  021000  106105                               ROLB    R5              ;SHIFT BIT IN R5
 4141  021002  001341                               BNE     64$             ;IF R5=0 THEN DONE
 4142  021004                                        ENDSEG
 4143  021004                                10000$:
 4144  021004  104405                               TRAP    C$ESEG
 4145  021006  012705  000001                        MOV     #1,R5           ;START WITH BIT 0
 4146  021012                                        BGNSEG
 4147  021012  104404                               TRAP    C$BSEG
 4148  021014                                67$:
 4149  021014  005105                               COM     R5
 4150  021016  010561  000004                        MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4151  021022                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4152  021026  122101                               122100!1                ;MOV DATA TO IBUS* REGISTER 1
 4153  021030                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4154  021034  021025                               21005!<1*20>            ;READ FROM IBUS* REGISTER 1
 4155  021036  010537  002612                        MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
 4156  021042  116104  000005                        MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 4157  021046  123704  002612                        CMPB    $GDDAT,R4               ;DATA CORRECT?
 4158  021052  001413                               BEQ     68$             ;BR IF YES
 4159  021054                                        ERROR   29              ;ERROR
 4160  021072  104455                               TRAP    C$ERDF
```

```
 4161  021074  000035                    .WORD   29
 4162  021076  006054                    .WORD   EM29
 4163  021100  011450                    .WORD   ERR29
 4164  021102                    68$:    ESCAPE  SEG
 4165  021102  104410                    TRAP    C$ESCAPE
 4166  021104  000012                    .WORD   10001$-.
 4167  021106  005105                    COM     R5              ;CHANGE TO FLOATING 1
 4168  021110  000241                    CLC                     ;CLEAR CARRY
 4169  021112  106105                    ROLB    R5              ;SHIFT BIT IN R5
 4170  021114  001337                    BNE     67$             ;IF R5=0 THEN DONE
 4171  021116                            ENDSEG
 4172  021116                    10001$:
 4173  021116  104405                    TRAP    C$ESEG
 4174  021120                    ENDTST
 4175  021120                    L10077:
 4176  021120  104401                    TRAP    C$ETST
```

```
021122                                  BADHEAD
                                        ;******************** TEST 24 ********************
                                        ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
                                        ;*FLOAT A 1 THOUGH IBUS REGISTER 2
                                        ;*FLOAT A 0 THROUGH IBUS REGISTER 2
021122                                  BADHEAD
                                        ;******************** TEST 24 ********************

021122                          BGNTST
021122                          T24::
021122                                  MSTCLR                  ;MASTER CLEAR COMM. MICRO-PROCCESSOR FAMILY
021126  012737  000002  002576          MOV     #2,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
021134  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
021140                                  MYINT
021144                                  BGNSEG
021144  104404                          TRAP    C$BSEG
021146                          64$:
021146  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
021152                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
021156  122102                          122100!2                ;MOV DATA TO IBUS* REGISTER 2
021160                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
021164  021045                          21005!<2*20>            ;READ FROM IBUS* REGISTER 2
021166  010537  002612                  MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
021172  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
021176  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
021202  001413                          BEQ     65$             ;BR IF YES
021204                                  ERROR   29              ;ERROR
021222  104455                          TRAP    C$ERDF
021224  000035                          .WORD   29
021226  006054                          .WORD   EM29
021230  011450                          .WORD   ERR29
021232                          65$:    ESCAPE  SEG
021232  104410                          TRAP    C$ESCAPE
021234  000010                          .WORD   10000$-.
021236  000241                          CLC                     ;CLEAR CARRY
021240  106105                          ROLB    R5              ;SHIFT BIT IN R5
021242  001341                          BNE     64$             ;IF R5=0 THEN DONE
021244                                  ENDSEG
021244                          10000$:
021244  104405                          TRAP    C$ESEG
021246  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
021252                                  BGNSEG
021252  104404                          TRAP    C$BSEG
021254                          67$:
021254  005105                          COM     R5
021256  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
021262                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
021266  122102                          122100!2                ;MOV DATA TO IBUS* REGISTER 2
021270                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
021274  021045                          21005!<2*20>            ;READ FROM IBUS* REGISTER 2
021276  010537  002612                  MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
021302  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
021306  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
021312  001413                          BEQ     68$             ;BR IF YES
021314                                  ERROR   29              ;ERROR
021332  104455                          TRAP    C$ERDF
```

```
4233  021334  000035                    .WORD   29
4234  021336  006054                    .WORD   EM29
4235  021340  011450                    .WORD   ERR29
4236  021342                    68$:    ESCAPE  SEG
4237  021342  104410                    TRAP    C$ESCAPE
4238  021344  000012                    .WORD   10001$-.
4239  021346  005105                    COM     R5              ;CHANGE TO FLOATING 1
4240  021350  000241                    CLC                     ;CLEAR CARRY
4241  021352  106105                    ROLB    R5              ;SHIFT BIT IN R5
4242  021354  001337                    BNE     67$             ;IF R5=0 THEN DONE
4243  021356                            ENDSEG
4244  021356                    10001$:
4245  021356  104405                    TRAP    C$ESEG
4246  021360                    ENDTST
4247  021360                    L10100:
4248  021360  104401                    TRAP    C$ETST
```

```
 4249  021362                                 BADHEAD
 4250                                         ;******************** TEST 25 **********************
 4251                                         ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
 4252                                         ;*FLOAT A 1 THOUGH IBUS REGISTER 3
 4253                                         ;*FLOAT A 0 THROUGH IBUS REGISTER 3
 4254  021362                                 BADHEAD
 4255                                         ;******************** TEST 25 **********************
 4256
 4257  021362                           BGNTST
 4258  021362                           T25::
 4259  021362                                 MSTCLR                  ;MASTER CLEAR COMM. MICRO-PROCCESSOR FAMILY
 4260  021366  012737  000003  002576         MOV     #3,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
 4261  021374  012705  000001                 MOV     #1,R5           ;START WITH BIT 0
 4262  021400                                 MYINT
 4263  021404                                 BGNSEG
 4264  021404  104404                         TRAP    C$BSEG
 4265  021406                           64$:
 4266  021406  010561  000004                 MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4267  021412                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4268  021416  122103                         122100!3                ;MOV DATA TO IBUS* REGISTER 3
 4269  021420                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4270  021424  021065                         21005!<3*20>            ;READ FROM IBUS* REGISTER 3
 4271  021426  010537  002612                 MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
 4272  021432  116104  000005                 MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 4273  021436  123704  002612                 CMPB    $GDDAT,R4               ;DATA CORRECT?
 4274  021442  001413                         BEQ     65$             ;BR IF YES
 4275  021444                                 ERROR   29              ;ERROR
 4276  021462  104455                         TRAP    C$ERDF
 4277  021464  000035                         .WORD   29
 4278  021466  006054                         .WORD   EM29
 4279  021470  011450                         .WORD   ERR29
 4280  021472                           65$:  ESCAPE  SEG
 4281  021472  104410                         TRAP    C$ESCAPE
 4282  021474  000010                         .WORD   10000$-.
 4283  021476  000241                         CLC                     ;CLEAR CARRY
 4284  021500  106105                         ROLB    R5              ;SHIFT BIT IN R5
 4285  021502  001341                         BNE     64$             ;IF R5=0 THEN DONE
 4286  021504                                 ENDSEG
 4287  021504                           10000$:
 4288  021504  104405                         TRAP    C$ESEG
 4289  021506  012705  000001                 MOV     #1,R5           ;START WITH BIT 0
 4290  021512                                 BGNSEG
 4291  021512  104404                         TRAP    C$BSEG
 4292  021514                           67$:
 4293  021514  005105                         COM     R5
 4294  021516  010561  000004                 MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4295  021522                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4296  021526  122103                         122100!3                ;MOV DATA TO IBUS* REGISTER 3
 4297  021530                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4298  021534  021065                         21005!<3*20>            ;READ FROM IBUS* REGISTER 3
 4299  021536  010537  002612                 MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
 4300  021542  116104  000005                 MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 4301  021546  123704  002612                 CMPB    $GDDAT,R4               ;DATA CORRECT?
 4302  021552  001413                         BEQ     68$             ;BR IF YES
 4303  021554                                 ERROR   29              ;ERROR
 4304  021572  104455                         TRAP    C$ERDF
```

```
 4305  021574  000035                    .WORD   29
 4306  021576  006054                    .WORD   EM29
 4307  021600  011450                    .WORD   ERR29
 4308  021602                    68$:    ESCAPE  SEG
 4309  021602  104410                    TRAP    C$ESCAPE
 4310  021604  000012                    .WORD   10001$-.
 4311  021606  005105                    COM     R5              ;CHANGE TO FLOATING 1
 4312  021610  000241                    CLC                     ;CLEAR CARRY
 4313  021612  106105                    ROLB    R5              ;SHIFT BIT IN R5
 4314  021614  001337                    BNE     67$             ;IF R5=0 THEN DONE
 4315  021616                            ENDSEG
 4316  021616                    10001$:
 4317  021616  104405                    TRAP    C$ESEG
 4318  021620                    ENDTST
 4319  021620                    L10101:
 4320  021620  104401                    TRAP    C$ETST
```

```
 4321  021622                                  BADHEAD
 4322                                          ;********************* TEST 26 *********************
 4323                                          ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
 4324                                          ;*FLOAT A 1 THOUGH IBUS REGISTER 4
 4325                                          ;*FLOAT A 0 THROUGH IBUS REGISTER 4
 4326  021622                                  BADHEAD
 4327                                          ;********************* TEST 26 *********************
 4328
 4329  021622                          BGNTST
 4330  021622                          T26::
 4331  021622                                  MSTCLR                  ;MASTER CLEAR COMM. MICRO-PROCCESSOR FAMILY
 4332  021626  012737  000004  002576         MOV     #4,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
 4333  021634  012705  000001                 MOV     #1,R5           ;START WITH BIT 0
 4334  021640                                  MYINT
 4335  021644                                  BGNSEG
 4336  021644  104404                         TRAP    C$BSEG
 4337  021646                          64$:
 4338  021646  010561  000004                 MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4339  021652                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4340  021656  122104                         122100!4                ;MOV DATA TO IBUS* REGISTER 4
 4341  021660                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4342  021664  021105                         21005!<4*20>            ;READ FROM IBUS* REGISTER 4
 4343  021666  010537  002612                 MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
 4344  021672  116104  000005                 MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 4345  021676  123704  002612                 CMPB    $GDDAT,R4               ;DATA CORRECT?
 4346  021702  001413                         BEQ     65$             ;BR IF YES
 4347  021704                                  ERROR   29              ;ERROR
 4348  021722  104455                         TRAP    C$ERDF
 4349  021724  000035                         .WORD   29
 4350  021726  006054                         .WORD   EM29
 4351  021730  011450                         .WORD   ERR29
 4352  021732                          65$:    ESCAPE  SEG
 4353  021732  104410                         TRAP    C$ESCAPE
 4354  021734  000010                         .WORD   10000$-.
 4355  021736  000241                         CLC                     ;CLEAR CARRY
 4356  021740  106105                         ROLB    R5              ;SHIFT BIT IN R5
 4357  021742  001341                         BNE     64$             ;IF R5=0 THEN DONE
 4358  021744                                  ENDSEG
 4359  021744                          10000$:
 4360  021744  104405                         TRAP    C$ESEG
 4361  021746  012705  000001                 MOV     #1,R5           ;START WITH BIT 0
 4362  021752                                  BGNSEG
 4363  021752  104404                         TRAP    C$BSEG
 4364  021754                          67$:
 4365  021754  005105                         COM     R5
 4366  021756  010561  000004                 MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
 4367  021762                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4368  021766  122104                         122100!4                ;MOV DATA TO IBUS* REGISTER 4
 4369  021770                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4370  021774  021105                         21005!<4*20>            ;READ FROM IBUS* REGISTER 4
 4371  021776  010537  002612                 MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
 4372  022002  116104  000005                 MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
 4373  022006  123704  002612                 CMPB    $GDDAT,R4               ;DATA CORRECT?
 4374  022012  001413                         BEQ     68$             ;BR IF YES
 4375  022014                                  ERROR   29              ;ERROR
 4376  022032  104455                         TRAP    C$ERDF
```

```
022034  000035                  .WORD   29
022036  006054                  .WORD   EM29
02[illegible]0  011450          .WORD   ERR29
022042                  68$:    ESCAPE  SEG
022042  104410                  TRAP    C$ESCAPE
022044  000012                  .WORD   10001$-.
022046  005105                  COM     R5              ;CHANGE TO FLOATING 1
022050  000241                  CLC                     ;CLEAR CARRY
022052  106105                  ROLB    R5              ;SHIFT BIT IN R5
022054  001337                  BNE     67$             ;IF R5=0 THEN DONE
022056                          ENDSEG
022056                  10001$:
022056  104405                  TRAP    C$ESEG
022060                  ENDTST
022060                  L10102:
022060  104401                  TRAP    C$ETST
```

```
                                        BADHEAD
                                        ;********************* TEST 27 *********************
                                        ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
                                        ;*FLOAT A 1 THOUGH IBUS REGISTER 5
                                        ;*FLOAT A 0 THROUGH IBUS REGISTER 5
                                        BADHEAD
                                        ;********************* TEST 27 *********************

                                BGNTST
                                T27::
                                        MSTCLR                  ;MASTER CLEAR COMM. MICRO-PROCCESSOR FAMILY
022066  012737  000005  002576          MOV     #5,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
022074  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
                                        MYINT
                                        BGNSEG
022104  104404                         TRAP    C$BSEG
                                64$:
022106  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
                                        122100!5                ;MOV DATA TO IBUS* REGISTER 5
                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
                                        21005!<5*20>            ;READ FROM IBUS* REGISTER 5
022126  010537  002612                  MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
022132  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
022136  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
022142  001413                         BEQ     65$             ;BR IF YES
                                        ERROR   29              ;ERROR
022162  104455                         TRAP    C$ERDF
022164  000035                         .WORD   29
022166  006054                         .WORD   EM29
022170  011450                         .WORD   ERR29
                                65$:    ESCAPE  SEG
022172  104410                         TRAP    C$ESCAPE
022174  000010                         .WORD   10000$-.
022176  000241                         CLC                     ;CLEAR CARRY
022200  106105                         ROLB    R5              ;SHIFT BIT IN R5
022202  001341                         BNE     64$             ;IF R5=0 THEN DONE
                                        ENDSEG
                                10000$:
022204  104405                         TRAP    C$ESEG
022206  012705  000001                  MOV     #1,R5           ;START WITH BIT 0
                                        BGNSEG
022212  104404                         TRAP    C$BSEG
                                67$:
022214  005105                         COM     R5
022216  010561  000004                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
022222                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
022226  122105                         122100!5                ;MOV DATA TO IBUS* REGISTER 5
022230                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
022234  021125                         21005!<5*20>            ;READ FROM IBUS* REGISTER 5
022236  010537  002612                  MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
022242  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
022246  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
022252  001413                         BEQ     68$             ;BR IF YES
022254                                  ERROR   29              ;ERROR
022272  104455                         TRAP    C$ERDF
```

```
4449  022274  000035                    .WORD   29
4450  022276  006054                    .WORD   EM29
4451  022300  011450                    .WORD   ERR29
4452  022302                    68$:    ESCAPE  SEG
4453  022302  104410                    TRAP    C$ESCAPE
4454  022304  000012                    .WORD   10001$-.
4455  022306  005105                    COM     R5              ;CHANGE TO FLOATING 1
4456  022310  000241                    CLC                     ;CLEAR CARRY
4457  022312  106105                    ROLB    R5              ;SHIFT BIT IN R5
4458  022314  001337                    BNE     67$             ;IF R5=0 THEN DONE
4459  022316                            ENDSEG
4460  022316                    10001$:
4461  022316  104405                    TRAP    C$ESEG
4462  022320                    ENDTST
4463  022320                    L10103:
4464  022320  104401                    TRAP    C$ETST
```

```
022322                                  BADHEAD
                                        ;******************** TEST 28 *********************
                                        ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
                                        ;*FLOAT A 1 THOUGH IBUS REGISTER 6
                                        ;*FLOAT A 0 THROUGH IBUS REGISTER 6
022322                                  BADHEAD
                                        ;******************** TEST 28 *********************

022322                          BGNTST
022322                          T28::
022322                                  MSTCLR                          ;MASTER CLEAR COMM. MICRO-PROCCESSOR FAMILY
022326  012737  000006  002576          MOV     #6,MR0                  ;SAVE REGISTER ADDRESS FOR TYPEOUT
022334  012705  000001                  MOV     #1,R5                   ;START WITH BIT 0
022340                                  MYINT
022344                                  BGNSEG
022344  104404                          TRAP    C$BSEG
022346                          64$:
022346  010561  000004                  MOV     R5,4(R1)                ;PUT PATTERN INTO PORT4
022352                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
022356  122106                          122100!6                        ;MOV DATA TO IBUS* REGISTER 6
022360                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
022364  021145                          21005!<6*20>                    ;READ FROM IBUS* REGISTER 6
022366  010537  002612                  MOV     R5,$GDDAT                       ;PUT EXPECTED IN $GDDAT
022372  116104  000005                  MOVB    5(R1),R4                ;PUT "FOUND" INTO R4
022376  123704  002612                  CMPB    $GDDAT,R4                       ;DATA CORRECT?
022402  001413                          BEQ     65$                     ;BR IF YES
022404                                  ERROR   29                      ;ERROR
022422  104455                          TRAP    C$ERDF
022424  000035                          .WORD   29
022426  006054                          .WORD   EM29
022430  011450                          .WORD   ERR29
022432                          65$:    ESCAPE  SEG
022432  104410                          TRAP    C$ESCAPE
022434  000010                          .WORD   10000$-.
022436  000241                          CLC                             ;CLEAR CARRY
022440  106105                          ROLB    R5                      ;SHIFT BIT IN R5
022442  001341                          BNE     64$                     ;IF R5=0 THEN DONE
022444                                  ENDSEG
022444                          10000$:
022444  104405                          TRAP    C$ESEG
022446  012705  000001                  MOV     #1,R5                   ;START WITH BIT 0
022452                                  BGNSEG
022452  104404                          TRAP    C$BSEG
022454                          67$:
022454  005105                          COM     R5
022456  010561  000004                  MOV     R5,4(R1)                ;PUT PATTERN INTO PORT4
022462                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
022466  122106                          122100!6                        ;MOV DATA TO IBUS* REGISTER 6
022470                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
022474  021145                          21005!<6*20>                    ;READ FROM IBUS* REGISTER 6
022476  010537  002612                  MOV     R5,$GDDAT                       ;PUT EXPECTED IN $GDDAT
022502  116104  000005                  MOVB    5(R1),R4                ;PUT "FOUND" INTO R4
022506  123704  002612                  CMPB    $GDDAT,R4                       ;DATA CORRECT?
022512  001413                          BEQ     68$                     ;BR IF YES
022514                                  ERROR   29                      ;ERROR
022532  104455                          TRAP    C$ERDF
```

```
 4521  022534  000035                .WORD   29
 4522  022536  006054                .WORD   EM29
 4523  022540  011450                .WORD   ERR29
 4524  022542                68$:    ESCAPE  SEG
 4525  022542  104410                TRAP    C$ESCAPE
 4526  022544  000012                .WORD   10001$-.
 4527  022546  005105                COM     R5              ;CHANGE TO FLOATING 1
 4528  022550  000241                CLC                     ;CLEAR CARRY
 4529  022552  106105                ROLB    R5              ;SHIFT BIT IN R5
 4530  022554  001337                BNE     67$             ;IF R5=0 THEN DONE
 4531  022556                        ENDSEG
 4532  022556                10001$:
 4533  022556  104405                TRAP    C$ESEG
 4534  022560                ENDTST
 4535  022560                L10104:
 4536  022560  104401                TRAP    C$ETST
```

```
                                        BADHEAD
                                        ;******************** TEST 29 **********************
                                        ;*MICRO PROCEOR IBUS* REGISTER WRITE/READ TEST
                                        ;*FLOAT A 1 THOUGH IBUS* REGISTER 7
                                        ;*FLOAT A 0 THROUGH IBUS* REGISTER 7
                                        BADHEAD
                                        ;******************** TEST 29 **********************

                                BGNTST
                                T29::
                                        MSTCLR                  ;MASTER CLEAR COMM. MICRO-PROCCESSOR FAMILY
022566  012737  000007  002576         MOV     #7,MR0          ;SAVE REGISTER ADDRESS FOR TYPEOUT
022574  012705  000001                 MOV     #1,R5           ;START WITH BIT 0
                                        MYINT
                                        BGNSEG
022604  104404                         TRAP    C$BSEG
                                64$:
022606  010561  000004                 MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
022616  122107                         122100!7                ;MOV DATA TO IBUS* REGISTER 7
                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
022624  021165                         21005!<7*20>            ;READ FROM IBUS* REGISTER 7
022626  010537  002612                 MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
022632  116104  000005                 MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
022636  123704  002612                 CMPB    $GDDAT,R4               ;DATA CORRECT?
022642  001413                         BEQ     65$             ;BR IF YES
                                        ERROR   29              ;ERROR
022662  104455                         TRAP    C$ERDF
022664  000035                         .WORD   29
022666  006054                         .WORD   EM29
022670  011450                         .WORD   ERR29
                                65$:    ESCAPE  SEG
022672  104410                         TRAP    C$ESCAPE
022674  000010                         .WORD   10000$-.
022676  000241                         CLC                     ;CLEAR CARRY
022700  106105                         ROLB    R5              ;SHIFT BIT IN R5
022702  001341                         BNE     64$             ;IF R5=0 THEN DONE
                                        ENDSEG
                                10000$:
022706  012705  000001                 TRAP    C$ESEG
                                        MOV     #1,R5           ;START WITH BIT 0
                                        BGNSEG
022714  005105                         TRAP    C$BSEG
                                67$:
022716  010561  000004                 COM     R5
022722                                  MOV     R5,4(R1)        ;PUT PATTERN INTO PORT4
022726  122107                         ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
022730                                  122100!7                ;MOV DATA TO IBUS* REGISTER 7
022734  021165                         ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
022736  010537  002612                 21005!<7*20>            ;READ FROM IBUS* REGISTER 7
022742  116104  000005                 MOV     R5,$GDDAT               ;PUT EXPECTED IN $GDDAT
022746  123704  002612                 MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
022752  001413                         CMPB    $GDDAT,R4               ;DATA CORRECT?
022754                                  BEQ     68$             ;BR IF YES
022772  104455                         ERROR   29              ;ERROR
                                        TRAP    C$ERDF
```

```
4593  022774  000035                .WORD   29
4594  022776  006054                .WORD   EM29
4595  023000  011450                .WORD   ERR29
4596  023002                68$:    ESCAPE  SEG
4597  023002  104410                TRAP    C$ESCAPE
4598  023004  000012                .WORD   10001$-.
4599  023006  005105                COM     R5              ;CHANGE TO FLOATING 1
4600  023010  000241                CLC                     ;CLEAR CARRY
4601  023012  106105                ROLB    R5              ;SHIFT BIT IN R5
4602  023014  001337                BNE     67$             ;IF R5=0 THEN DONE
4603  023016                        ENDSEG
4604  023016                10001$:
4605  023016  104405                TRAP    C$ESEG
4606  023020                ENDTST
4607  023020                L10105:
4608  023020  104401                TRAP    C$ETST
```

```
 4609  023022                                  BADHEAD
 4610                                          ;********** TEST 30 **********
 4611                                          ;*MICRO PROCESSOR IBUS DUAL ADDRESS TEST
 4612                                          ;*WRITE ALL IBUS REGISTERS WITH INCREMENTING PATTERN
 4613                                          ;*READ ALL IBUS REGISTERS TO VERIFY CORRECT ADDRESSING
 4614  023022                                  BADHEAD
 4615                                          ;********** TEST 30 **********
 4616
 4617  023022                          BGNTST
 4618  023022                          T30::
 4619  023022                                  MSTCLR                  ;MASTER CLEAR COMM. MICRO-PROCCESSOR FAMILY
 4620  023026  012705  000001                 MOV     #1,R5           ;START WITH A ONE
 4621  023032  005002                         CLR     R2              ;R2 CONTAINS ADDRESS OF REGISTER
 4622  023034                                  MYINT
 4623  023040                                  BGNSEG
 4624  023040  104404                         TRAP    C$BSEG
 4625  023042  010203                  1$:     MOV     R2,R3           ;R3=REGISTER ADDRESS
 4626  023044  010561  000004                 MOV     R5,4(R1)        ;WRITE DATA TO PORT4
 4627  023050  042737  000017  023066         BIC     #17,5$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 4628  023056  050337  023066                 BIS     R3,5$           ;ADD ADDRESS TO INSTRUCTION
 4629  023062                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4630  023066  122100                  5$:     122100                  ;MOVE DATA TO IBUS REGISTER
 4631  023070  006303                         ASL     R3              ;SHIFT ADDRESS
 4632  023072  006303                         ASL     R3              ;4 TIMES TO GET
 4633  023074  006303                         ASL     R3              ;IT TO BITS 4-7
 4634  023076  006303                         ASL     R3              ;OF NEXT INSTRUCTION
 4635  023100  042737  000360  023116         BIC     #360,6$         ;CLEAR ADDRESS FIELD
 4636  023106  050337  023116                 BIS     R3,6$           ;ADD ADDRESS TO INSTRUCTION
 4637  023112                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4638  023116  021005                  6$:     21005                   ;READ FROM IBUS REGISTER
 4639  023120  010537  002612                 MOV     R5,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
 4640  023124  116104  000005                 MOVB    5(R1),R4        ;PUT "FOUND" IN R4
 4641  023130  123704  002612                 CMPB    $GDDAT,R4               ;IS DATA CORRECT?
 4642  023134  001413                         BEQ     2$              ;BR IF YES
 4643  023136                                  ERROR   29              ;DATA ERROR
 4644  023154  104455                         TRAP    C$ERDF
 4645  023156  000035                         .WORD   29
 4646  023160  006054                         .WORD   EM29
 4647  023162  011450                         .WORD   ERR29
 4648  023164                          2$:     ESCAPE  SEG
 4649  023164  104410                         TRAP    C$ESCAPE
 4650  023166  000014                         .WORD   10000$-.
 4651  023170  005205                         INC     R5              ;INCREMENT PATTERN
 4652  023172  005202                         INC     R2              ;INCREMENT REGISTER ADDRESS
 4653  023174  022702  000010                 CMP     #7+1,R2         ;LAST ADDRESS DONE?
 4654  023200  001320                         BNE     1$              ;BR IF NO
 4655  023202                                  ENDSEG
 4656  023202                          10000$:
 4657  023202  104405                         TRAP    C$ESEG
 4658  023204  012705  000001                 MOV     #1,R5           ;RESTART PATTERN TO 1
 4659  023210  005002                         CLR     R2              ;RESTART AT ADDRESS 0
 4660  023212                                  BGNSEG
 4661  023212  104404                         TRAP    C$BSEG
 4662  023214  005003                         CLR     R3              ;RESTART AT ADDRESS 0
 4663  023216  042737  000360  023234  3$:     BIC     #360,7$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
 4664  023224  050337  023234                 BIS     R3,7$           ;ADD ADDRESS TO INSTRUCTION
```

```
023230                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
023234  021005                  7$:     21005                   ;READ FROM IBUS REGISTER
023236  010537  002612                  MOV     R5,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
023242  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" IN $GDDAT
023246  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
023252  001413                          BEQ     4$              ;BR IF YES
023254                                  ERROR   30              ;DUAL ADDRESSING ERROR
023272  104455                          TRAP    C$ERDF
023274  000036                          .WORD   30
023276  004622                          .WORD   EM30
023300  011562                          .WORD   ERR30
023302                          4$:     ESCAPE  SEG
023302  104410                          TRAP    C$ESCAPE
023304  000020                          .WORD   10001$-.
023306  005205                          INC     R5              ;INCREMENT PATTERN
023310  005202                          INC     R2              ;NEXT ADDRESS
023312  062703  000020                  ADD     #20,R3          ;ADD 1 TO ADDRESS IN R3(SHIFTED 4 TIMES)
023316  022702  000010                  CMP     #7+1,R2         ;LAST ADDRESS DONE?
023322  001335                          BNE     3$              ;BR IF NO
023324                                  ENDSEG
023324                          10001$:
023324  104405                          TRAP    C$ESEG
023326                          ENDTST
023326                          L10106:
023326  104401                          TRAP    C$ETST
```

```
 4690  023330                                  BADHEAD
 4691                                          ;******************** TEST 31 **********************
 4692                                          ;*IBUS*0-10,1-11,3-13 DUAL ADDRESS TEST
 4693                                          ;*READ IBUS*10,11,13 AND SAVE VALUES
 4694                                          ;*WRITE IBUS*0,1,3 WITH THE RESPECTIVE COMPLEMENTS OF THESE VALUES
 4695                                          ;*CHECK IBUS*10,11,13 FOR ANY CHANGES
 4696  023330                                  BADHEAD
 4697                                          ;******************** TEST 31 **********************
 4698
 4699  023330                            BGNTST
 4700  023330                            T31::
 4701  023330                                  MSTCLR                          ; CLEAR M8206
 4702  023334                                  MYINT
 4703  023340  005037  002632                 CLR     SIBS10                  ; CLEAR SAVE REGISTERS
 4704  023344  005037  002634                 CLR     SIBS11
 4705  023350  005037  002640                 CLR     SIBS13
 4706  023354                                  ROMCLK
 4707  023360  121205                         121005!<10*20>                  ; READ IBUS* 10
 4708  023362  116137  000005  002632         MOVB    5(R1),SIBS10            ; SAVE IT
 4709  023370                                  ROMCLK
 4710  023374  121225                         121005!<11*20>                  ; READ IBUS* 11
 4711  023376  116137  000005  002634         MOVB    5(R1),SIBS11            ; SAVE IT
 4712  023404                                  ROMCLK
 4713  023410  121265                         121005!<13*20>                  ; READ IBUS* 13
 4714  023412  116137  000005  002640         MOVB    5(R1),SIBS13            ; SAVVE IT
 4715  023420  013705  002632           1$:   MOV     SIBS10,R5               ; GET DATA
 4716  023424  005105                         COM     R5                      ; MAKE OPPOSITE
 4717  023426  110561  000004                 MOVB    R5,4(R1)                ; LOAD 0
 4718  023432                                  ROMCLK
 4719  023436  121100                         121100
 4720  023440  012705  000070                 MOV     #70,R5                  ; LEGAL BITS
 4721  023444  110561  000004                 MOVB    R5,4(R1)                ; LOAD 1
 4722  023450                                  ROMCLK
 4723  023454  121101                         121101
 4724  023456  013705  002640                 MOV     SIBS13,R5               ; GET DATA
 4725  023462  005105                         COM     R5                      ; MAKE OPPOSITE
 4726  023464  110561  000004                 MOVB    R5,4(R1)                ; LOAD 3
 4727  023470                                  ROMCLK
 4728  023474  121103                         121103
 4729  023476  012737  000010  002576   2$:   MOV     #10,MR0                 ; CHECK 10
 4730  023504  013737  002632  002612         MOV     SIBS10,$GDDAT           ; EXPECTED
 4731  023512                                  ROMCLK
 4732  023516  121205                         121005!<10*20>                  ; READ 10
 4733  023520  116104  000005                 MOVB    5(R1),R4                ; GET DATA
 4734  023524  123704  002612                 CMPB    $GDDAT,R4               ; CHECK IT
 4735  023530  001413                         BEQ     3$                      ; SKIP IF OK
 4736  023532                                  ERROR   27
 4737  023550  104455                         TRAP    C$ERDF
 4738  023552  000033                         .WORD   27
 4739  023554  005774                         .WORD   EM27
 4740  023556  011230                         .WORD   ERR27
 4741  023560  012737  000011  002576   3$:   MOV     #11,MR0                 ; CHECK 11
 4742  023566  013737  002634  002612         MOV     SIBS11,$GDDAT           ; EXPECTED
 4743  023574                                  ROMCLK
 4744  023600  121225                         121005!<11*20>
 4745  023602  116104  000005                 MOVB    5(R1),R4                ; GET DATA
```

```
 4746  023606  123704  002612                    CMPB    $GDDAT,R4               ; CHECK IT
 4747  023612  001413                            BEQ     4$                      ; SKIP IF OK
 4748  023614                                    ERROR   27
 4749  023632  104455                            TRAP    C$ERDF
 4750  023634  000033                            .WORD   27
 4751  023636  005774                            .WORD   EM27
 4752  023640  011230                            .WORD   ERR27
 4753  023642  012737  000013  002576  4$:       MOV     #13,MR0                 ; CHECK 13
 4754  023650  013737  002640  002612            MOV     SIBS13,$GDDAT           ; EXPECTED
 4755  023656                                    ROMCLK
 4756  023662  121265                            121005!<13*20>                  ; READ 13
 4757  023664  116104  000005                    MOVB    5(R1),R4                ; GET DATA
 4758  023670  123704  002612                    CMPB    $GDDAT,R4               ; CHECK IT
 4759  023674  001413                            BEQ     5$                      ; SKIP IF OK
 4760  023676                                    ERROR   27
 4761  023714  104455                            TRAP    C$ERDF
 4762  023716  000033                            .WORD   27
 4763  023720  005774                            .WORD   EM27
 4764  023722  011230                            .WORD   ERR27
 4765  023724                          5$:       ENDTST
 4766  023724                          L10107:
 4767  023724  104401                            TRAP    C$ETST
```

```
023726                                          BADHEAD
                                                ;******************** TEST 32 ********************
                                                ;*TEST THE DELAY ON PASS 1 AND REPORT ITS VALUE
023726                                          BADHEAD
                                                ;******************** TEST 32 ********************

023726                                  BGNTST
023726                                  T32::
023726                                          MSTCLR                          ; CLEAR M8206
023732                                          MYINT
023736  005037  002630                          CLR     SCLK                    ; CLEAR FLAG
023742  012761  000020  000004                  MOV     #20,4(R1)               ; HIT TIMER
023750                                          ROMCLK
023754  121111                                  121111
023756                                          DELAY   1                       ; WAIT 100 USEC
023756  012727  000001                          MOV     #1,(PC)+
023762  000000                                  .WORD   0
023764  013727  002116                          MOV     L$DLY,(PC)+
023770  000000                                  .WORD   0
023772  005367  177772                          DEC     -6(PC)
023776  001375                                  BNE     .-4
024000  005367  177756                          DEC     -22(PC)
024004  001367                                  BNE     .-20
024006                                          ROMCLK
024012  121225                                  121005!<11*20>                  ; READ IBUS* 11
024014  132761  000020  000005                  BITB    #20,5(R1)               ; CHECK CLOCK
024022  001140                                  BNE     2$                      ; SKIP IF SET
024024                                          DELAY   250.                    ; WAIT 25 MSEC
024024  012727  000372                          MOV     #250.,(PC)+
024030  000000                                  .WORD   0
024032  013727  002116                          MOV     L$DLY,(PC)+
024036  000000                                  .WORD   0
024040  005367  177772                          DEC     -6(PC)
024044  001375                                  BNE     .-4
024046  005367  177756                          DEC     -22(PC)
024052  001367                                  BNE     .-20
024054                                          DELAY   250.                    ; WAIT 25 MSEC
024054  012727  000372                          MOV     #250.,(PC)+
024060  000000                                  .WORD   0
024062  013727  002116                          MOV     L$DLY,(PC)+
024066  000000                                  .WORD   0
024070  005367  177772                          DEC     -6(PC)
024074  001375                                  BNE     .-4
024076  005367  177756                          DEC     -22(PC)
024102  001367                                  BNE     .-20
024104                                          DELAY   250.                    ; WAIT 25 MSEC
024104  012727  000372                          MOV     #250.,(PC)+
024110  000000                                  .WORD   0
024112  013727  002116                          MOV     L$DLY,(PC)+
024116  000000                                  .WORD   0
024120  005367  177772                          DEC     -6(PC)
024124  001375                                  BNE     .-4
024126  005367  177756                          DEC     -22(PC)
024132  001367                                  BNE     .-20
024134                                          DELAY   250.                    ; WAIT 25 MSEC
024134  012727  000372                          MOV     #250.,(PC)+
```

```
4824  024140  000000                        .WORD   0
4825  024142  013727  002116                MOV     L$DLY,(PC)+
4826  024146  000000                        .WORD   0
4827  024150  005367  177772                DEC     -6(PC)
4828  024154  001375                        BNE     .-4
4829  024156  005367  177756                DEC     -22(PC)
4830  024162  001367                        BNE     .-20
4831  024164                                DELAY   250.                    ; WAIT 25 MSEC
4832  024164  012727  000372                MOV     #250.,(PC)+
4833  024170  000000                        .WORD   0
4834  024172  013727  002116                MOV     L$DLY,(PC)+
4835  024176  000000                        .WORD   0
4836  024200  005367  177772                DEC     -6(PC)
4837  024204  001375                        BNE     .-4
4838  024206  005367  177756                DEC     -22(PC)
4839  024212  001367                        BNE     .-20
4840  024214                                DELAY   250.                    ; WAIT 25 MSEC
4841  024214  012727  000372                MOV     #250.,(PC)+
4842  024220  000000                        .WORD   0
4843  024222  013727  002116                MOV     L$DLY,(PC)+
4844  024226  000000                        .WORD   0
4845  024230  005367  177772                DEC     -6(PC)
4846  024234  001375                        BNE     .-4
4847  024236  005367  177756                DEC     -22(PC)
4848  024242  001367                        BNE     .-20
4849  024244                                ROMCLK
4850  024250  121225                        121005!<11*20>                  ; READ IBUS* 11
4851  024252  132761  000020  000005        BITB    #20,5(R1)               ; CHECK CLOCK
4852  024260  001404                        BEQ     1$                      ; NOT SET
4853  024262  012737  000001  002630        MOV     #1,SCLK                 ; SET SLOW FLAG
4854  024270  000415                        BR      2$                      ; LONG DELAY
4855  024272                        1$:     ERROR   27                      ; CLOCK DID NOT SET
4856  024310  104455                        TRAP    C$ERDF
4857  024312  000033                        .WORD   27
4858  024314  005774                        .WORD   EM27
4859  024316  011230                        .WORD   ERR27
4860  024320                                ESCAPE  TST
4861  024320  104410                        TRAP    C$ESCAPE
4862  024322  000066                        .WORD   L10110-.
4863  024324  005737  002604        2$:     TST     PONE                    ; SKIP IF NOT FIRST PASS
4864  024330  001027                        BNE     4$
4865  024332  012737  177777  002604        MOV     #-1,PONE                ; SET SWITCH
4866  024340  005737  002630                TST     SCLK                    ; FAST OR SLOW?
4867  024344  001011                        BNE     3$                      ; SKIP IF SLOW
4868  024346                                PRINTB  #FMSG
4869  024346  012746  004407                MOV     #FMSG,-(SP)
4870  024352  012746  000001                MOV     #1,-(SP)
4871  024356  010600                        MOV     SP,R0
4872  024360  104414                        TRAP    C$PNTB
4873  024362  062706  000004                ADD     #4,SP
4874  024366  000410                        BR      4$
4875  024370                        3$:     PRINTB  #SMSG
4876  024370  012746  004457                MOV     #SMSG,-(SP)
4877  024374  012746  000001                MOV     #1,-(SP)
4878  024400  010600                        MOV     SP,R0
4879  024402  104414                        TRAP    C$PNTB
```

```
 4880  024404  062706  000004                ADD     #4,SP
 4881  024410                          4$:   ENDTST
 4882  024410                          L10110:
 4883  024410  104401                        TRAP    C$ETST
```

```
 4884  024412                                  BADHEAD
 4885                                          ;******************** TEST 33 ********************
 4886                                          ;*MICRO PROCESSOR BR REGISTER TEST
 4887                                          ;*FLOAT A 1 THOUGH THE BR
 4888                                          ;*FLOAT A 0 THOUGH THE BR
 4889  024412                                  BADHEAD
 4890                                          ;******************** TEST 33 ********************
 4891
 4892  024412                          BGNTST
 4893  024412                          T33::
 4894
 4895  024412                                  MSTCLR                  ;R1 CONTAINS BASE COMM. MICRO-PROCCESSOR FAMILY ADDRESS
 4896  024416  012705  000001                  MOV     #1,R5           ;MASTER CLEAR COMM. MICRO-PROCCESSOR FAMILY
 4897  024422                                  MYINT                   ;START PATTERN WITH BITO
 4898  024426                                  BGNSEG
 4899  024426  104404                          TRAP    C$BSEG
 4900  024430                          64$:
 4901  024430  010561  000004                  MOV     R5,4(R1)        ;WRITE PATTERN IN PORT4
 4902  024434                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4903  024440  120500                          120500                  ;MOVE DATA TO THE BR REGISTER
 4904  024442                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4905  024446  061225                          061225                  ;MOVE BR TO PORT 5
 4906  024450  010537  002612                  MOV     R5,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
 4907  024454  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" IN R4
 4908  024460  123704  002612                  CMPB    $GDDAT,R4               ;IS DATA CORRECT?
 4909  024464  001413                          BEQ     65$             ;BR IF YES
 4910  024466                                  ERROR   3               ;DATA ERROR
 4911  024504  104455                          TRAP    C$ERDF
 4912  024506  000003                          .WORD   3
 4913  024510  004665                          .WORD   EM3
 4914  024512  006624                          .WORD   ERR3
 4915  024514                          65$:    ESCAPE  SEG
 4916  024514  104410                          TRAP    C$ESCAPE
 4917  024516  000010                          .WORD   10000$-.
 4918  024520  000241                          CLC                     ;CLEAR CARRY
 4919  024522  106105                          ROLB    R5              ;SHIFT BIT IN R5
 4920  024524  001341                          BNE     64$             ;DONE IF R5=0
 4921  024526                                  ENDSEG
 4922  024526                          10000$:
 4923  024526  104405                          TRAP    C$ESEG
 4924  024530  012705  000001                  MOV     #1,R5           ;START PATTERN WITH BITO
 4925  024534                          69$:
 4926  024534                                  BGNSEG
 4927  024534  104404                          TRAP    C$BSEG
 4928  024536                          67$:
 4929  024536  005105                          COM     R5
 4930  024540  010561  000004                  MOV     R5,4(R1)        ;WRITE PATTERN IN PORT4
 4931  024544                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4932  024550  120500                          120500                  ;MOVE DATA TO THE BR REGISTER
 4933  024552                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
 4934  024556  061225                          061225                  ;MOVE BR TO PORT 5
 4935  024560  010537  002612                  MOV     R5,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
 4936  024564  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" IN $GDDAT
 4937  024570  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
 4938  024574  001413                          BEQ     68$             ;BR IF YES
 4939  024576                                  ERROR   3               ;DATA ERROR
```

```
 4940  024614  104455                        TRAP    C$ERDF
 4941  024616  000003                        .WORD   3
 4942  024620  004665                        .WORD   EM3
 4943  024622  006624                        .WORD   ERR3
 4944  024624                        68$:    ESCAPE  SEG
 4945  024624  104410                        TRAP    C$ESCAPE
 4946  024626  000070                        .WORD   10001$-.
 4947  024630  052711  040000                BIS     #40000,(R1)     ;SET MASTER CLEAR
 4948  024634  042711  040000                BIC     #40000,(R1)     ;CLEAR IT
 4949  024640                                ROMCLK                  ;PUT BR IN PORT5
 4950  024644  061225                        061225
 4951  024646  116104  000005                MOVB    5(R1),R4        ;READ IT
 4952  024652  001415                        BEQ     70$             ;IF ZERO, OK
 4953  024654  005005                        CLR     R5
 4954  024656                                ERROR 3                 ;MASTER CLEAR
 4955  024674  104455                        TRAP    C$ERDF
 4956  024676  000003                        .WORD   3
 4957  024700  004665                        .WORD   EM3
 4958  024702  006624                        .WORD   ERR3
 4959  024704                                CKLOOP
 4960  024704  104406                        TRAP    C$CLP1
 4961
 4962  024706                        70$:                            ;FAILED TO CLEAR
 4963  024706  005105                                                ;BRG
 4964  024710  000241                        COM     R5              ;CHANGE BACK TO A ONE
 4965  024712  106105                        CLC                     ;CLEAR CARRY
 4966  024714  001310                        ROLB    R5              ;SHIFT BIT IN R5
 4967  024716                                BNE     67$             ;DONE IF R5=0
 4968  024716                                ENDSEG
 4969  024716  104405                10001$: TRAP    C$ESEG
 4970  024720                        ENDTST
 4971  024720                        L10111:
 4972  024720  104401                        TRAP    C$ETST
 4973
 4974  024722                                BADHEAD
 4975                                        ;********************* TEST 34 *********************
 4976                                        ;*SCRATCH PAD TEST
 4977                                        ;*FLOAT A 1 THOUGH EACH SCRATCH PAD LOCATION
 4978                                        ;*FLOAT A 0 THOUGH EACH SCRATCH PAD LOCATION
 4979  024722                                BADHEAD
 4980                                        ;********************* TEST 34 *********************
 4981
 4982  024722                        BGNTST
 4983  024722                        T34::
 4984  024722                                MYINT
 4985  024726                                MSTCLR                  ;MASTER CLEAR COMM. MICRO-PROCCESSOR FAMILY
 4986  024732  005002                        CLR     R2              ;START AT ADDRESS ZERO
 4987  024734  012705  000001                MOV     #1,R5           ;START WITH BIT0
 4988  024740                                BGNSUB
 4989  024740                        T34.1:
 4990  024740  104402                        TRAP    C$BSUB
 4991  024742                        1$:     BGNSEG
 4992  024742  104404                        TRAP    C$BSEG
 4993  024744  042737  000017  024766  64$:  BIC     #17,65$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
 4994  024752  050237  024766                BIS     R2,65$          ;ADD ADDRESS TO INSTRUCTION
 4995  024756  010561  000004                MOV     R5,4(R1)        ;WRITE PATTERN IN PORT4
```

```
024762                                      ROMCLK                      ;NEXT WORD IS INSTRUCTION,  BBN
024766  123100                      65$:    123100                      ;WRITE SCRATCH PAD(ADDRESS IN R2)
024770  042737  000017  025006              BIC     #17,66$             ;CLEAR ADDRESS FIELD OF INSTRUCTION
024776  050237  025006                      BIS     R2,66$              ;ADD ADDRESS TO INSTRUCTION
025002                                      ROMCLK                      ;NEXT WORD IS INSTRUCTION,  BBN
025006  040600                      66$:    040600                      ;MOVE SP TO BR
025010                                      ROMCLK                      ;NEXT WORD IS INSTRUCTION,  BBN
025014  061225                              061225                      ;MOVE BR TO PORT5
025016  010537  002612                      MOV     R5,$GDDAT                   ;PUT "EXPECTED" IN $GDDAT
025022  116104  000005                      MOVB    5(R1),R4            ;PUT "FOUND" IN R4
025026  123704  002612                      CMPB    $GDDAT,R4                   ;DATA CORRECT
025032  001413                              BEQ     67$                 ;BR IF YES
025034                                      ERROR   4                   ;DATA ERROR
025052  104455                              TRAP    C$ERDF
025054  000004                              .WORD   4
025056  004713                              .WORD   EM4
025060  006732                              .WORD   ERR4
025062                              67$:    ESCAPE  SEG
025062  104410                              TRAP    C$ESCAPE
025064  000010                              .WORD   10000$-.
025066  000241                              CLC                         ;CLEAR CARRY
025070  106105                              ROLB    R5                  ;SHIFT BIT IN R5
025072  001324                              BNE     64$                 ;DONE IF R5=0
025074                                      ENDSEG
025074                              10000$:
025074  104405                              TRAP    C$ESEG
025076  012705  000001                      MOV     #1,R5               ;START WITH BIT0
025102                                      BGNSEG
025102  104404                              TRAP    C$BSEG

025104  005105                      73$:    COM     R5                  ;CHANGE TO FLOATING ZERO
025106  042737  000017  025130      69$:    BIC     #17,70$             ;CLEAR ADDRESS FIELD OF INSTRUCTION
025114  050237  025130                      BIS     R2,70$              ;ADD ADDRESS TO INSTRUCTION
025120  010561  000004                      MOV     R5,4(R1)            ;WRITE PATTERN IN PORT4
025124                                      ROMCLK                      ;NEXT WORD IS INSTRUCTION,  BBN
025130  123100                      70$:    123100                      ;WRITE SCRATCH PAD(ADDRESS IN R2)
025132  042737  000017  025150              BIC     #17,71$             ;CLEAR ADDRESS FIELD OF INSTRUCTION
025140  050237  025150                      BIS     R2,71$              ;ADD ADDRESS TO INSTRUCTION
025144                                      ROMCLK                      ;NEXT WORD IS INSTRUCTION,  BBN
025150  040600                      71$:    040600                      ;MOVE SP TO BR
025152                                      ROMCLK                      ;NEXT WORD IS INSTRUCTION,  BBN
025156  061225                              061225                      ;MOVE BR TO PORT5
025160  010537  002612                      MOV     R5,$GDDAT                   ;PUT "EXPECTED" IN $GDDAT
025164  116104  000005                      MOVB    5(R1),R4            ;PUT "FOUND" IN $GDDAT
025170  123704  002612                      CMPB    $GDDAT,R4                   ;DATA CORRECT?
025174  001413                              BEQ     72$                 ;BR IF YES
025176                                      ERROR   4                   ;DATA ERROR
025214  104455                              TRAP    C$ERDF
025216  000004                              .WORD   4
025220  004713                              .WORD   EM4
025222  006732                              .WORD   ERR4
025224                              72$:    ESCAPE  TST
025224  104410                              TRAP    C$ESCAPE
025226  000032                              .WORD   L10112-.
025230  005105                              COM     R5                  ;CHANGE BACK TO A ONE
025232  000241                              CLC                         ;CLEAR CARRY
```

```
5052  025234  106105                        ROLB    R5              ;SHIFT BIT IN R5
5053  025236  001322                        BNE     73$             ;DONE IF R5=0
5054  025240                          ENDSEG
5055  025240                          10001$:
5056  025240  104405                        TRAP    C$ESEG
5057  025242  012705  000001                MOV     #1,R5           ;RESTART AT BIT 0
5058  025246  005202                        INC     R2              ;NEXT SP ADDRESS
5059  025250  022702  000020                CMP     #20,R2          ;LAST ADDRESS?
5060  025254  001232                        BNE     1$              ;BR IF NO
5061  025256                                ENDSUB
5062  025256                          L10113:
5063  025256  104403                        TRAP    C$ESUB
5064  025260                          ENDTST
5065  025260                          L10112:
5066  025260  104401                        TRAP    C$ETST
5067
5068  025262                                BADHEAD
5069                                        ;********************* TEST 35 *********************
5070                                        ;*SCRATCH PAD DUAL ADDRESSING TEST
5071                                        ;*WRITE AN INCREMENTING PATTERN IN ALL SP LOCATIONS
5072                                        ;*READ ALL SP LOCATIONS TO VERIFY CORRECT ADDRESSING
5073  025262                                BADHEAD
5074                                        ;********************* TEST 35 *********************
5075
5076  025262                          BGNTST
5077  025262                          T35::
5078  025262                                MSTCLR                  ;MASTER CLEAR COMM. MICRO-PROCCESSOR FAMILY
5079  025266  012705  000001                MOV     #1,R5           ;START WITH A 1
5080  025272  005003                        CLR     R3              ;ADDRESS 0
5081  025274                                MYINT
5082  025300                                BGNSEG
5083  025300  104404                        TRAP    C$BSEG
5084  025302  010302                1$:     MOV     R3,R2                   ;MOVE ADDRESS TO R2
5085  025304  042737  000017  025326        BIC     #17,2$                  ;CLEAR ADDRESS FIELD
5086  025312  050237  025326                BIS     R2,2$                   ;ADD ADDRESS TO INSTRUCTION
5087  025316  010561  000004                MOV     R5,4(R1)                ;WRITE PATTERN IN PORT4
5088  025322                                ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
5089  025326  123100                2$:     123100                          ;WRITE SP(ADDRESS IN R2)
5090  025330  042737  000017  025346        BIC     #17,3$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
```

```
5091  025336  050237  025346          BIS     R2,3$           ;ADD ADDRESS TO INSTRUCTION
5092  025342                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
```

```
5093  025346  060600                  3$:     60600                   ;MOVE SP TO BR
5094  025350                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5095  025354  061225                          61225                   ;MOVE BR TO PORT5
5096  025356  010537  002612                  MOV     R5,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
5097  025362  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" IN R4
5098  025366  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT
5099  025372  001413                          BEQ     4$              ;BR IF YES
5100  025374                                  ERROR   4               ;DATA ERROR
5101  025412  104455                          TRAP    C$ERDF
5102  025414  000004                          .WORD   4
5103  025416  004713                          .WORD   EM4
5104  025420  006732                          .WORD   ERR4
5105  025422                          4$:     ESCAPE  SEG
5106  025422  104410                          TRAP    C$ESCAPE
5107  025424  000014                          .WORD   10000$-.
5108  025426  005205                          INC     R5              ;INCREMENT PATTERN
5109  025430  005203                          INC     R3              ;NEXT ADDRESS
5110  025432  022703  000020                  CMP     #20,R3          ;LAST ADDRESS DONE?
5111  025436  001321                          BNE     1$              ;BR IF NO
5112  025440                                  ENDSEG
5113  025440                          10000$:
5114  025440  104405                          TRAP    C$ESEG
5115  025442  012705  000001                  MOV     #1,R5           ;RESTART PATTERN AT 1
5116  025446  005003                          CLR     R3              ;RESTART AT ADDRESS ZERO
5117  025450                                  BGNSEG
5118  025450  104404                          TRAP    C$BSEG
5119  025452  010302                  5$:     MOV     R3,R2           ;PUT ADDRESS IN R2
5120  025454  042737  000017  025472  69$:    BIC     #17,6$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
5121  025462  050237  025472                  BIS     R2,6$           ;ADD ADDRESS TO INSTRUCTION
5122  025466                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5123  025472  060600                  6$:     60600                   ;MOV SP TO BR
5124  025474                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION;  BBN
5125  025500  061225                          61225                   ;MOV BR TO PORT5
5126  025502  010537  002612                  MOV     R5,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
5127  025506  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" IN $GDDAT
5128  025512  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
5129  025516  001413                          BEQ     7$              ;BR IF YES
5130  025520                                  ERROR   5               ;SP ADDRESSING ERROR
5131  025536  104455                          TRAP    C$ERDF
5132  025540  000005                          .WORD   5
5133  025542  004741                          .WORD   EM5
5134  025544  007044                          .WORD   ERR5
5135  025546                          7$:     ESCAPE  SEG
5136  025546  104410                          TRAP    C$ESCAPE
5137  025550  000014                          .WORD   10001$-.
5138  025552  005205                          INC     R5              ;INCREMENT PATTERN
5139  025554  005203                          INC     R3              ;NEXT ADDRESS
5140  025556  022703  000020                  CMP     #20,R3          ;LAST ADDRESS DONE?
5141  025562  001333                          BNE     5$              ;BR IF NO
5142  025564                                  ENDSEG
5143  025564                          10001$:
5144  025564  104405                          TRAP    C$ESEG
5145  025566                          ENDTST
5146  025566                          L10114:
5147  025566  104401                          TRAP    C$ETST
5148
```

```
5149  025570                                  BADHEAD
5150                                          ;******************** TEST 36 *********************
5151                                          ;*INTERRUPT TEST
5152                                          ;*TEST THAT DEVICE CAN INTERRUPT TO VECTOR A
5153  025570                                  BADHEAD
5154                                          ;******************** TEST 36 *********************
5155
5156  025570                            BGNTST
5157  025570                            T36::
5158  025570                                  MYINT
5159  025574  004737  004146                 JSR     PC,PRDEL            ; WAIT FOR ANY PRINTOUTS TO CLEAR TERMINAL
5160  025600  000005                          RESET                       ;BUS RESET
5161  025602  005011                          CLR     (R1)                ;CLEAR RUN
5162  025604  004537  003604                 JSR     R5,SETVEC           ;SET UP VECTORS
5163  025610  025730                          3$                          ;XX0
5164  025612  025702                          2$                          ;XX4
5165  025614  000340  000340                 .WORD   340,340             ;LEVEL 7
5166  025620                            1$:   SETPRI  #PRI07              ;PS = LEVEL 7
5167  025620  012700  000340                 MOV     #PRI07,R0
5168  025624  104441                          TRAP    C$SPRI
5169  025626  012761  000200  000004         MOV     #200,4(R1)          ;WRITE PORT4
5170  025634                                  ROMCLK                      ;NEXT WORD IS INSTRUCTION,  BBN
5171  025640  121111                          121111                      ;SET BR RQ IN IBUS* REG 11
5172  025642                                  SETPRI  #PRI00              ;ALLOW INTERRUPT
5173  025642  012700  000000                 MOV     #PRI00,R0
5174  025646  104441                          TRAP    C$SPRI
5175  025650  000240                          NOP
5176  025652                                  ERROR   31                  ;NO INTERRUPT
5177  025670  104455                          TRAP    C$ERDF
5178  025672  000037                          .WORD   31
5179  025674  005561                          .WORD   EM31
5180  025676  011670                          .WORD   ERR31
5181  025700  000415                          BR      4$
5182  025702                            2$:   ERROR   32                  ;WRONG VECTOR
5183  025720  104455                          TRAP    C$ERDF
5184  025722  000040                          .WORD   32
5185  025724  005610                          .WORD   EM32
5186  025726  011746                          .WORD   ERR32
5187  025730  062706  000004            3$:   ADD     #4,SP     ;RESET STACK
5188  025734                            4$:
5189  025734                            ENDTST
5190  025734                            L10115:
5191  025734  104401                          TRAP    C$ETST
5192
5193  025736                                  BADHEAD
5194                                          ;******************** TEST 37 *********************
5195                                          ;*INTERRUPT TEST
5196                                          ;*TEST THAT DEVICE CAN INTERRUPT TO VECTOR B
5197  025736                                  BADHEAD
5198                                          ;******************** TEST 37 *********************
5199
5200  025736                            BGNTST
5201  025736                            T37::
5202  025736                                  MYINT
5203  025742                                  MSTCLR                      ;MASTER CLEAR M8200,4,6,7
5204  025746  004537  003604                 JSR     R5,SETVEC           ;SET UP VECTORS
```

```
5205  025752  026044                          2$                      ;XX0
5206  025754  026072                          3$                      ;XX4
5207  025756  000340  000340                  .WORD   340,340         ;LEVEL 7
5208  025762                          1$:     SETPRI  #PRI07          ;PS = LEVEL 7
5209  025762  012700  000340                  MOV     #PRI07,R0
5210  025766  104441                          TRAP    C$SPRI
5211  025770  012761  000300  000004          MOV     #300,4(R1)      ;WRITE PORT4
5212  025776                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5213  026002  121111                          121111                  ;SET BR RQ IN IBUS* REG 11
5214  026004                                  SETPRI  #PRI00          ;ALLOW INTERRUPT
5215  026004  012700  000000                  MOV     #PRI00,R0
5216  026010  104441                          TRAP    C$SPRI
5217  026012  000240                          NOP
5218  026014                                  ERROR   31              ;NO INTERRUPT
5219  026032  104455                          TRAP    C$ERDF
5220  026034  000037                          .WORD   31
5221  026036  005561                          .WORD   EM31
5222  026040  011670                          .WORD   ERR31
5223  026042  000415                          BR      4$
5224  026044                          2$:     ERROR   32              ;WRONG VECTOR
5225  026062  104455                          TRAP    C$ERDF
5226  026064  000040                          .WORD   32
5227  026066  005610                          .WORD   EM32
5228  026070  011746                          .WORD   ERR32
5229  026072  062706  000004          3$:     ADD     #4,SP   ;RESET STACK
5230  026076                          4$:
5231  026076                          ENDTST
5232  026076                          L10116:
5233  026076  104401                          TRAP    C$ETST
5234
5235  026100                                  BADHEAD
5236                                          ;******************** TEST 38 **********************
5237                                          ;*PRIORITY INTERRUPT TEST
5238                                          ;*SET PS TO ALL BR LEVELS EQUAL OR GREATER THAN
5239                                          ;*THE M8200,4,6,7 LEVEL, VERIFY THAT COMM. MICRO-PROCESSOR FAMILY DOES NOT INTER
5240  026100                                  BADHEAD
5241                                          ;******************** TEST 38 **********************
5242
5243  026100                          BGNTST
5244  026100                          T38::
5245  026100                                  MYINT
5246  026104                                  MSTCLR                  ;MASTER CLEAR M8200,4,6,7
5247  026110  012704  000340                  MOV     #340,R4         ;PUT LEVEL 7 IN R2
5248  026114                                  SETPRI  R4              ;SET PRIORITY TO 7
5249  026114  010400                          MOV     R4,R0
5250  026116  104441                          TRAP    C$SPRI
5251  026120  013705  002666                  MOV     STAT1,R5        ;GET BR LEVEL OF M8200,4,6,7
5252  026124  006205                          ASR     R5              ;SHIFT R5 4 TIMES
5253  026126  006205                          ASR     R5              ;TO GET PROPER LEVEL
5254  026130  006205                          ASR     R5
5255  026132  006205                          ASR     R5
5256  026134  042705  177437                  BIC     #177437,R5      ;CLEAR UNWANTED BITS
5257  026140  010537  002612                  MOV     R5,$GDDAT
5258  026144  004537  003604                  JSR     R5,SETVEC       ;SET UP VECTORS
5259  026150  026214                          2$                      ;A VECTOR
5260  026152  026214                          2$                      ;B VECTOR
```

```
5261  026154  000340  000340
5262  026160  012761  000200  000004  4$:     MOV     #200,4(R1)      ;LOAD PORT4
5263  026166                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5264  026172  121111                          121111                  ;SET BR REQUEST
5265  026174                          5$:     SETPRI  R4              ;PUT LEVEL IN R2 IN PS
5266  026174  010400                          MOV     R4,R0
5267  026176  104441                          TRAP    C$SPRI
5268  026200  000240                          NOP
5269  026202  020504                          CMP     R5,R4           ;IS PRESENT PS LEVEL = TO M8200,4,6,7 LEVEL
5270  026204  001417                          BEQ     1$              ;BR IF YES
5271  026206  162704  000040                 SUB     #40,R4          ;NO GET NEXT LOWER LEVEL IN R2
5272  026212  000770                          BR      5$              ;AND CONTINUE WITH TEST
5273  026214                          2$:     ERROR   33              ;ERROR UNEXPECTED INTERRUPT
5274  026232  104455                          TRAP    C$ERDF
5275  026234  000041                          .WORD   33
5276  026236  005647                          .WORD   EM33
5277  026240  012024                          .WORD   ERR33
5278  026242  000002                          RTI
5279  026244                          1$:     MSTCLR
5280  026250                                  ENDTST
5281  026250                          L10117:
5282  026250  104401                          TRAP    C$ETST
5283
5284  026252                                  BADHEAD
5285                                          ;********************* TEST 39 *********************
5286                                          ;*PRIORITY INTERRUPT TESTS
5287                                          ;*SET PS TO ALL BR LEVELS LESS THAN THE M8200,4,6,7 LEVEL
5288                                          ;*VERIFY THAT M8200,4,6,7 WILL INTERRUPT
5289  026252                                  BADHEAD
5290                                          ;********************* TEST 39 *********************
5291
5292  026252                          BGNTST
5293  026252                          T39::
5294  026252                                  MYINT
5295  026256                                  MSTCLR
5296  026262  012704  000340                 MOV     #340,R4         ;MASTER CLEAR M8200,4,6,7
5297  026266                                  SETPRI  R4              ;PUT LEVEL 7 IN R2
5298  026266  010400                          MOV     R4,R0           ;SET PRIORITY TO 7
5299  026270  104441                          TRAP    C$SPRI
5300  026272  013705  002666                 MOV     STAT1,R5        ;GET BR LEVEL OF M8200,4,6,7
5301  026276  006205                          ASR     R5              ;SHIFT R5 4 TIMES
5302  026300  006205                          ASR     R5              ;TO GET PROPER LEVEL
5303  026302  006205                          ASR     R5
5304  026304  006205                          ASR     R5
5305  026306  042705  177437                 BIC     #177437,R5      ;CLEAR UNWANTED BITS
5306  026312  010502                          MOV     R5,R2           ;PUT M8200,4,6,7 LEVEL IN R2
5307  026314  162702  000040                 SUB     #40,R2          ;GET NEXT LOWER LEVEL IN R2
5308  026320  004537  003604                 JSR     R5,SETVEC       ;SET UP VECTORS
5309  026324  026406                          2$                      ;A VECTOR
5310  026326  026414                          3$                      ;B VECTOR
5311  026330  000340  000340                 .WORD   340,340         ;PRIORITY 7
5312  026334  012761  000200  000004  4$:     MOV     #200,4(R1)      ;LOAD PORT4
5313  026342                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5314  026346  121111                          121111                  ;SET BR REQUEST
5315  026350                          5$:     SETPRI  R2              ;PUT LEVEL IN R2 IN PS
5316  026350  010200                          MOV     R2,R0
```

```
5317  026352  104441                          TRAP    C$SPRI
5318  026354  000240                          NOP
5319  026356                                  ERROR   31              ;ERROR, NO INTERRUPT
5320  026374  104455                          TRAP    C$ERDF
5321  026376  000037                          .WORD   31
5322  026400  005561                          .WORD   EM31
5323  026402  011670                          .WORD   ERR31
5324  026404  000421                  6$:     BR      1$
5325  026406  012716  026404          2$:     MOV     #6$,(SP)        ;SET UP FOR RTI
5326  026412  000002                          RTI
5327  026414                          3$:     ERROR   32              ;ERROR, WRONG VECTOR
5328  026432  104455                          TRAP    C$ERDF
5329  026434  000040                          .WORD   32
5330  026436  005610                          .WORD   EM32
5331  026440  011746                          .WORD   ERR32
5332  026442  012716  026450                  MOV     #1$,(SP)        ;SET UP FOR RTI
5333  026446  000002                          RTI
5334  026450                          1$:     MSTCLR
5335  026454                                  ENDTST
5336  026454                          L10120:
5337  026454  104401                          TRAP    C$ETST
5338
5339  026456                                  BADHEAD
5340                                          ;********************* TEST 40 *********************
5341                                          ;*NPR TEST
5342                                          ;*TEST OF DATO, 1 WORD FROM UPROC TO 11 MEMORY
5343  026456                                  BADHEAD
5344                                          ;********************* TEST 40 *********************
5345
5346  026456                          BGNTST
5347  026456                          T40::
5348  026456  004737  004146                  JSR     PC,PRDEL        ; WAIT FOR PRINTOUTS
5349  026462  000005                          RESET                   ;BUS RESET
5350
5351  026464                                  MYINT
5352  026470  005011                          CLR     (R1)            ;CLEAR RUN
5353  026472  005061  000004                  CLR     4(R1)           ;CLR PORT4
5354  026476  004537  003626                  JSR     R5,NPRSET       ;SET UP IBUS REG 0-7
5355  026502  000000                          0                       ;IN DATA
5356  026504  177777                          -1                      ;OUT DATA
5357  026506  026626                          3$                      ;IN BA
5358  026510  026624                          2$                      ;OUT BA
5359  026512  005037  026624                  CLR     2$              ;CLEAR 2$
5360  026516  005061  000004                  CLR     4(R1)           ;CLEAR PORT 4
5361  026522                                  ROMCLK                  ;NOW MOVE TO IBUS*<11>
5362  026526  121111                          121111
5363  026530  012761  000021  000004          MOV     #21,4(R1)       ;WRITE PORT4
5364  026536                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5365  026542  121110                          121110                  ;SET NPR BITS IN IBUS* REG 10
5366  026544  000240                          NOP
5367  026546  012737  177777  002612          MOV     #-1,$GDDAT              ;PUT "EXPECTED" IN $GDDAT
5368  026554  013704  026624                  MOV     2$,R4           ;PUT "FOUND" IN R4
5369  026560  023704  002612                  CMP     $GDDAT,R4               ;DATA CORRECT?
5370  026564  001415                          BEQ     4$              ;BR IF YES
5371  026566                                  ERROR   11              ;ERROR NPR FAILED
5372  026604  104455                          TRAP    C$ERDF
```

```
5373  026606  000013                          .WORD   11
5374  026610  005123                          .WORD   EM11
5375  026612  007514                          .WORD   ERR11
5376  026614                                  ESCAPE  TST
5377  026614  104410                          TRAP    C$ESCAPE
5378  026616  000012                          .WORD   L10121-.
5379  026620                          4$:     EXIT    TST
5380  026620  104432                          TRAP    C$EXIT
5381  026622  000006                          .WORD   L10121-.
5382  026624  000000                  2$:     0                       ;OUT BA
5383  026626  000000                  3$:     0                       ;IN BA
5384  026630                          ENDTST
5385  026630                          L10121:
5386  026630  104401                          TRAP    C$ETST
5387
5388  026632                                  BADHEAD
5389                                          ;********************* TEST 41 *********************
5390                                          ;*NPR TEST
5391                                          ;*TEST OF DATI, 1 WORD FROM 11 MEMORY TO UPROC
5392  026632                                  BADHEAD
5393                                          ;********************* TEST 41 *********************
5394
5395  026632                          BGNTST
5396  026632                          T41::
5397  026632                                  MYINT
5398  026636                                  MSTCLR                  ;MASTER CLEAR M8200,4,6,7
5399  026642  005061  000004                 CLR     4(R1)           ;CLR PORT4
5400  026646  004537  003626                 JSR     R5,NPRSET       ;SET UP IBUS REG 0-7
5401  026652  000000                          0                       ;IN DATA
5402  026654  177777                          -1                      ;OUT DATA
5403  026656  027002                          3$                      ;IN BA
5404  026660  027000                          2$                      ;OUT BA
5405  026662  012737  177777  027002          MOV     #-1,3$          ;PUT DATA IN 3$
5406  026670  012761  000001  000004          MOV     #1,4(R1)        ;WRITE PORT4
5407  026676                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5408  026702  121110                          121110                  ;SET NPR BITS IN IBUS* REG 11
5409  026704  000240                          NOP
5410  026706  012737  177777  002612          MOV     #-1,$GDDAT              ;PUT ''EXPECTED'' IN $GDDAT
5411  026714                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5412  026720  021004                          021004                  ;MOVE IN DATA LOW BYTE TO PORT4
5413  026722                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5414  026726  021025                          021025                  ;MOVE IN DATA HIGH BYTE TO PORT5
5415  026730  016104  000004                 MOV     4(R1),R4        ;PUT ''FOUND'' IN R4
5416  026734  023704  002612                 CMP     $GDDAT,R4               ;DATA CORRECT?
5417  026740  001415                          BEQ     4$              ;BR IF YES
5418  026742                                  ERROR   11              ;ERROR NPR FAILED
5419  026760  104455                          TRAP    C$ERDF
5420  026762  000013                          .WORD   11
5421  026764  005123                          .WORD   EM11
5422  026766  007514                          .WORD   ERR11
5423  026770                                  ESCAPE  TST
5424  026770  104410                          TRAP    C$ESCAPE
5425  026772  000012                          .WORD   L10122-.
5426  026774                          4$:     EXIT    TST
5427  026774  104432                          TRAP    C$EXIT
5428  026776  000006                          .WORD   L10122-.
```

```
5429  027000  000000                  2$:     0                       ;OUT BA
5430  027002  000000                  3$:     0                       ;IN BA
5431  027004                          ENDTST
5432  027004                          L10122:
5433  027004  104401                          TRAP    C$ETST
5434
5435  027006                          BADHEAD
5436                                  ;********************* TEST 42 *********************
5437                                  ;*NPR TEST
5438                                  ;*TEST OF DATOB, 1 BYTE FROM UPROC TO 11 MEMORY
5439  027006                          BADHEAD
5440                                  ;********************* TEST 42 *********************
5441
5442  027006                          BGNTST
5443  027006                          T42::
5444  027006                          MYINT
5445  027012                          MSTCLR                          ;MASTER CLEAR M8200,4,6,7
5446  027016  005061  000004         CLR     4(R1)                   ;CLR PORT4
5447  027022  004537  003626         JSR     R5,NPRSET               ;SET UP IBUS REG 0-7
5448  027026  000000                  0                               ;IN DATA
5449  027030  177777                  -1                              ;OUT DATA
5450  027032  027152                  3$                              ;IN BA
5451  027034  027151                  2$+1                            ;OUT BA
5452  027036  005037  027150         CLR     2$                      ;CLEAR 2$
5453  027042  005061  000004         CLR     4(R1)                   ;CLEAR PORT 4
5454  027046                          ROMCLK                          ;NOW MOVE IT TO IBUS*<11>
5455  027052  121111                  121111
5456  027054  012761  000221  000004 MOV     #221,4(R1)              ;WRITE PORT4
5457  027062                          ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
5458  027066  121110                  121110                          ;SET NPR BITS IN IBUS* REG 11
5459  027070  000240                  NOP
5460  027072  012737  177400  002612 MOV     #177400,$GDDAT          ;PUT "EXPECTED" IN $GDDAT
5461  027100  013704  027150         MOV     2$,R4                   ;PUT "FOUND" IN R4
5462  027104  023704  002612         CMP     $GDDAT,R4                       ;DATA CORRECT?
5463  027110  001415                  BEQ     4$                      ;BR IF YES
5464  027112                          ERROR   11                      ;ERROR NPR FAILED
5465  027130  104455                  TRAP    C$ERDF
5466  027132  000013                  .WORD   11
5467  027134  005123                  .WORD   EM11
5468  027136  007514                  .WORD   ERR11
5469  027140                          ESCAPE  TST
5470  027140  104410                  TRAP    C$ESCAPE
5471  027142  000012                  .WORD   L10123-.
5472  027144                  4$:     EXIT    TST
5473  027144  104432                  TRAP    C$EXIT
5474  027146  000006                  .WORD   L10123-.
5475  027150  000000          2$:     0                               ;OUT BA
5476  027152  000000          3$:     0                               ;IN BA
5477  027154                          ENDTST
5478  027154                          L10123:
5479  027154  104401                  TRAP    C$ETST
5480
5481  027156                          BADHEAD
5482                                  ;********************* TEST 43 *********************
5483                                  ;*TEST OF EA BITS 16 AND 17
5484                                  ;*DO A DATO TO AN ADDRESS USING OUT BA BITS 16 AND 17
```

```
5485
5486  027156                                    ;*VERIFY CORRECT RESULTS
5487                                            BADHEAD
5488                                            ;********************* TEST 43 **********************
5489  027156                              BGNTST
5490  027156                              T43::
5491  027156                                    MSTCLR                  ;MASTER CLEAR M8200,4,6,7
5492  027162                                    MYINT
5493  027166  013737  002714  027214            MOV     KMP06,1$        ;USE SEL4 FOR ADDRESS
5494  027174  013737  002714  027212            MOV     KMP06,2$        ;USE SEL4 FOR ADDRESS
5495  027202  004537  003626                    JSR     R5,NPRSET       ;LOAD BA AND DATA
5496  027206  000000                            0                       ;IN DATA
5497  027210  125252                            125252                  ;OUT DATA
5498  027212  000000                      2$:   0                       ;IN BA
5499  027214  000000                      1$:   0                       ;OUT BA
5500  027216  012761  000014  000004            MOV     #14,4(R1)       ;LOAD SEL 4 WITH CUT BA16 AND 17
5501  027224                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION.  BBN
5502  027230  121111                            121111                  ;SET OUTBA 16 AND 17
5503  027232  012761  000021  000004            MOV     #21,4(R1)       ;LOAD SEL4
5504  027240  012711  003000                    MOV     #BIT9!BIT10,(R1)
5505  027244  012761  121110  000006            MOV     #121110,6(R1)   ;PUT INSTRUCTION IN SEL6
5506  027252  052711  000400                    BIS     #BIT8,(R1)      ;CLOCK IT!
5507  027256  000240                            NOP                     ;WAIT FOR NPR
5508  027260  012737  121110  002612            MOV     #121110,$GDDAT  ;PUT "EXPECTED" IN $GDDAT
5509  027266  000240                            NOP
5510  027270  000240                            NOP
5511                                                                    ;OK,LISTEN UP!EXPLAINATION TIME.
5512                                                                    ;
5513                                                                    ;ON THE NPR OUT,THE DATA ENDED UP
5514                                                                    ;IN THE IBUS(NOT IBUS*) SENCE SEL A
5515                                                                    ;WAS ONLY SELECTED IN THE NPR CYCLE.
5516                                                                    ;THAT IS,WE DIDN'T REALLY DO AN NPR TO
5517                                                                    ;PORT 6,THE NPR OUT REALLY ENDED UP IN
5518                                                                    ;OUT DATA LOW,AND OUT DATA HIGH
5519                                                                    ;(IBUS <2> AND IBUS <3>).
5520
5521                                                                    ;WHAT WE'RE DOING NEXT IS READING IBUS 2&3
5522                                                                    ;TO SEE IF THE DATA GOT XFERRED CORRECTLY.
5523  027272                                    ROMCLK
5524  027276  021044                            021044
5525  027300                                    ROMCLK                  ;READ IBUS <2> PUT IN PORT 4
5526  027304  021065                            021065
5527  027306  016104  000004                    MOV     4(R1),R4        ;READ IBUS <3> PUT IN PORT 5
5528  027312  023704  002612                    CMP     $GDDAT,R4       ;PUT "FOUND" IN R4
5529  027316  001413                            BEQ     3$                      ;CORRECT RESULTS?
5530  027320                                    ERROR   11              ;BR IF YES
5531  027336  104455                            TRAP    C$ERDF          ;ERROR BA 16 AND 17 FAILED
5532  027340  000013                            .WORD   11
5533  027342  005123                            .WORD   EM11
5534  027344  007514                            .WORD   ERR11
5535  027346                              3$:
5536  027346                              ENDTST
5537  027346                              L10124:
5538  027346  104401                            TRAP    C$ETST
5539
5540  027350                                    BADHEAD
```

```
                                                  ;******************** TEST 44 ********************
                                                  ;*TEST OF EA BITS 16 AND 17
                                                  ;*DO A DATI USING IN BA BITS 16 AND 17
                                                  ;*VERIFY CORRECT RESULTS
                                                  ;*IN ORDER TO DO THIS TEST, WE WILL READ THE DATA FROM THE
                                                  ;*CONSOL TTY CSR IF ONE EXSITS
                                                  ;*IF NO CONSOL TTY CSR AT ADDRESS 177560, THIS TEST
                                                  ;*WILL BE SKIPPED
027350                                            BADHEAD
                                                  ;******************** TEST 44 ********************

027350                                    BGNTST
027350                                    T44::
027350                                            MYINT
027354                                            MSTCLR                      ;MASTER CLEAR M8200,4,6,7
027360  012737  027546  000004            STOP:   MOV     #TOUTT,4            ;SET UP FOR TRAP IN CASE IF NO
027366  012737  000340  000006                    MOV     #340,6  ;TTY AT ADDRESS 177560
027374  005737  177560                            TST     177560              ;ADDRESS THE TTY-TRAPS HERE IF NONE.
027400  012737  177560  027426                    MOV     #177560,1$          ;USE SEL4 FOR ADDRESS
027406  012737  177560  027424                    MOV     #177560,2$          ;USE SEL4 FOR ADDRESS
027414  004537  003626                            JSR     R5,NPRSET           ;LOAD BA AND DATA
027420  000000                                    0                           ;IN DATA
027422  125252                                    125252                      ;OUT DATA
027424  000000                            2$:     0                           ;IN BA
027426  000000                            1$:     0                           ;OUT BA
027430  012761  000015  000004                    MOV     #15,4(R1)
027436  012711  003000                            MOV     #BIT9!BIT10,(R1);SET CROMI AND CROMO!!
027442  012761  121110  000006                    MOV     #121110,6(R1)       ;PUT INSTR INTO SEL6   NW*
027450  052711  000400                            BIS     #BIT8,(R1)          ;CLOCK IT!
027454  000240                                    NOP                         ;WAIT FOR NPR
027456                                            ROMCLK                      ;NEXT WORD IS INSTRUCTION,  BBN
027462  021004                                    021004                      ;MOVE OUT DATA LB TO SEL4
027464                                            ROMCLK                      ;NEXT WORD IS INSTRUCTION,  BBN
027470  021025                                    021025                      ;MOVE OUT DATA HB TO SEL5
027472  016104  000004                            MOV     4(R1),R4            ;PUT "FOUND" IN R4
027476  013737  177560  002612                    MOV     177560,$GDDAT
027504  042737  000200  002612                    BIC     #200,$GDDAT
027512  023704  002612                            CMP     $GDDAT,R4                   ;CORRECT RESULTS?
027516  001415                                    BEQ     TOUTP               ;BR IF YES
027520                                            ERROR   11                  ;ERROR BA 16 AND 17 FAILED
027536  104455                                    TRAP    C$ERDF
027540  000013                                    .WORD   11
027542  005123                                    .WORD   EM11
027544  007514                                    .WORD   ERR11
027546                                    3$:
027546  062706  000004                    TOUTT:  ADD     #4,SP               ;UPDATE STACK POITNTER
027552  013737  002626  000006            TOUTP:  MOV     SAVE6,6             ;RESTORE TRAP VECTOR
027560  013737  002624  000004                    MOV     SAVE4,4
027566                                    ENDTST
027566                                    L10125:
027566  104401                                    TRAP    C$ETST

027570                                            BADHEAD
                                                  ;******************** TEST 45 ********************
```

```
 5595                                           ;*NPR NON-EXISTENT MEMORY TEST
 5596                                           ;*DO A DATO TO A NON-EXISTENT ADDRESS
 5597                                           ;*VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
 5598  027570                                   BADHEAD
 5599                                           ;********************* TEST 45 *********************
 5600
 5601  027570                           BGNTST
 5602  027570                           T45::
 5603  027570                                   MYINT
 5604  027574                                   MSTCLR                          ;MASTER CLEAR M8200,4,6,7
 5605  027600  004537  003626                   JSR     R5,NPRSET               ;LOAD IBUS REGISTERS 0-7
 5606  027604  000000                           0                               ;IN DATA
 5607  027606  000000                           0                               ;OUT DATA
 5608  027610  177320                           177320                          ;IN BA
 5609  027612  177320                           177320                          ;IN BA
 5610  027614  012761  000014  000004           MOV     #14,4(R1)               ;SET OUT BA BITS 16+17 IN PORT4
```

```
5611  027622                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5612  027626  121111                            121111                  ;SET OUTBA 16 AND 17
5613  027630  012761  000021  000004            MOV     #21,4(R1)       ;SET NPR REQUEST BITS IN PORT4
5614  027636                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5615  027642  121110                            121110                  ;MOV IBUS* 4 TO IBUS* 10
5616  027644  000240                            NOP
5617  027646                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5618  027652  121225                            121225                  ;MOV IBUS*11 TO IBUS*5
5619  027654  012737  000001  002612            MOV     #1,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
5620  027662  116104  000005                    MOVB    5(R1),R4        ;PUT "FOUND" IN R4
5621  027666  042704  177776                    BIC     #177776,R4      ;CLEAR UNWANTED BITS
5622  027672  023704  002612                    CMP     $GDDAT,R4               ;DATA CORRECT?
5623  027676  001413                            BEQ     1$              ;BR IF YES
5624  027700                                    ERROR   13              ;ERROR NON-EXISTENT MEM BIT FAILED TO SET
5625  027716  104455                            TRAP    C$ERDF
5626  027720  000015                            .WORD   13
5627  027722  005156                            .WORD   EM13
5628  027724  007730                            .WORD   ERR13
5629  027726                          1$:
5630  027726  152761  000100  000001            BISB    #100,1(R1)      ;SET MASTER CLEAR
5631  027734  142761  000100  000001            BICB    #100,1(R1)      ;CLEAR MASTER
5632  027742                                    ROMCLK                  ;MOV IBUS*11 TO
5633  027746  121225                            121225                  ;PORT5
5634  027750  005037  002612                    CLR     $GDDAT          ;EXPECT CLEAR
5635  027754  116104  000005                    MOVB    5(R1),R4        ;GET NPR REG
5636  027760  042704  177776                    BIC     #177776,R4      ;CLEAR JUNK
5637  027764  001413                            BEQ     2$              ;EXIT IF CLEAR
5638  027766                                    ERROR   13              ;NON-EXISTANT MEM
5639  030004  104455                            TRAP    C$ERDF
5640  030006  000015                            .WORD   13
5641  030010  005156                            .WORD   EM13
5642  030012  007730                            .WORD   ERR13
5643                                                                    ;BIT FAILED TO CLEAR
5644  030014                          2$:
5645  030014                          ENDTST
5646  030014                          L10126:
5647  030014  104401                            TRAP    C$ETST
5648
5649  030016                                    BADHEAD
5650                                            ;********************* TEST 46 *********************
5651                                            ;*NPR NON-EXISTENT MEMORY TEST
5652                                            ;*DO A DATI FROM A NON-EXISTENT ADDRESS
5653                                            ;*VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
5654  030016                                    BADHEAD
5655                                            ;********************* TEST 46 *********************
5656
5657  030016                          BGNTST
5658  030016                          T46::
5659  030016                                    MYINT
5660  030022                                    MSTCLR                  ;MASTER CLEAR M8200,4,6,7
5661  030026  004537  003626                    JSR     R5,NPRSET       ;LOAD IBUS REGISTERS 0-7
5662  030032  000000                            0                       ;IN DATA
5663  030034  000000                            0                       ;OUT DATA
5664  030036  177320                            177320                  ;IN BA
5665  030040  177320                            177320                  ;OUT BA
5666  030042  005061  000004                    CLR     4(R1)
```

```
5667  030046
5668  030052  121111                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
                                              121111                  ;CLEAR NON-EXISTENT BIT
5669  030054  012761  000015  000004          MOV     #15,4(R1)       ;SET NPR REQUEST BITS IN PORT4
5670  030062                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5671  030066  121110                          121110                  ;MOV IBUS* 4 TO IBUS* 10
5672  030070  000240                          NOP
5673  030072                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5674  030076  121225                          121225                  ;MOV IBUS*11 TO IBUS*5
5675  030100  012737  000001  002612          MOV     #1,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
5676  030106  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" IN R4
5677  030112  042704  177776                  BIC     #177776,R4      ;CLEAR UNWANTED BITS
5678  030116  023704  002612                  CMP     $GDDAT,R4               ;DATA CORRECT?
5679  030122  001413                          BEQ     1$              ;BR IF YES
5680  030124                                  ERROR   13              ;ERROR NON-EXISTENT MEM BIT FAILED TO SET
5681  030142  104455                          TRAP    C$ERDF
5682  030144  000015                          .WORD   13
5683  030146  005156                          .WORD   EM13
5684  030150  007730                          .WORD   ERR13
5685  030152                          1$:
5686  030152                          ENDTST
5687  030152                          L10127:
5688  030152  104401                          TRAP    C$ETST
5689
5690  030154                                  BADHEAD
5691                                          ;********************* TEST 47 *********************
5692                                          ;*NPR TEST
5693                                          ;*USING DATO, NPR A BINARY COUNT (0-377)
5694                                          ;*FROM MICRO-PROCESSOR TO ALL AVAILABLE MEMORY
5695  030154                                  BADHEAD
5696                                          ;********************* TEST 47 *********************
5697
5698  030154                          BGNTST
5699  030154                          T47::
5700  030154                                  MYINT
5701  030160                                  MSTCLR                  ;MASTER CLEAR M8200,4,6,7
5702  030164  005037  030372                  CLR     5$              ;START FLAG AT 0
5703  030170  005005                          CLR     R5              ;DATA
5704  030172  012702  041630                  MOV     #CORMAX,R2      ;ADDRESS
5705  030176                          1$:
5706  030176  010537  030226                  MOV     R5,2$           ;LOAD DATA
5707  030202  010237  030232                  MOV     R2,4$           ;LOAD BA
5708  030206  032702  000001                  BIT     #BIT0,R2        ;IS BA ODD?
5709  030212  001402                          BEQ     .+6             ;BR IF NO
5710  030214  000337  030226                  SWAB    2$              ;IF ODD PUT DATA IN HI-BYTE
5711  030220  004537  003626                  JSR     R5,NPRSET       ;LOAD NPR REGISTERS
5712  030224  000000                          0                       ;IN DATA
5713  030226  000000                  2$:     0                       ;OUT DATA
5714  030230  000000                          0                       ;IN BA
5715  030232  000000                  4$:     0                       ;OUT BA
5716  030234  105012                          CLRB    (R2)            ;CLEAR MEMORY LOCATION
5717  030236  012761  000221  000004          MOV     #221,4(R1)      ;LOAD PORT4
5718  030244                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5719  030250  121110                          121110                  ;DO THE NPR
5720  030252  000240                          NOP
5721  030254  010537  002612                  MOV     R5,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
5722  030260  111204                          MOVB    (R2),R4 ;PUT "FOUND" IN R4
```

```
5723  030262  123704  002612          CMPB    $GDDAT,R4               ;IS DATA CORRECT?
5724  030266  001413                  BEQ     3$              ;BR IF YES
5725  030270                          ERROR   11              ;ERROR, DATA INCORRECT
5726  030306  104455                  TRAP    C$ERDF
5727  030310  000013                  .WORD   11
5728  030312  005123                  .WORD   EM11
5729  030314  007514                  .WORD   ERR11
5730  030316                  3$:     ESCAPE  TST
5731  030316  104410                  TRAP    C$ESCAPE
5732  030320  000054                  .WORD   L10130-.
5733  030322  005205                  INC     R5              ;NEXT CHARACTER
5734  030324  042705  177400          BIC     #177400,R5      ;USE ONLY LOW BYTE
5735  030330  005737  030372          TST     5$              ;HAS MAX MEMORY BEEN REACHED YET?
5736  030334  001402                  BEQ     6$              ;BR IF NO
5737  030336  005705                  TST     R5              ;DONE PATTERN?
5738  030340  001412                  BEQ     7$              ;BR IF YES
5739  030342  005202                  INC     R2              ;INC BA
5740  030344  023702  002556  6$:     CMP     MEMLIM,R2       ;REACHED MEMORY LIMIT YET?
5741  030350  001312                  BNE     1$              ;BR IF NOT
5742  030352  012702  041630          MOV     #CORMAX,R2      ;RESTART BA AT FIRST ADDRESS
5743  030356  012737  177777  030372  MOV     #-1,5$          ;SET FLAG TO END TEST AT END OF DATA PATTERN
5744  030364  000704                  BR 1$                   ;CONTINUE
5745  030366                  7$:
5746  030366                          EXIT    TST
5747  030366  104432                  TRAP    C$EXIT
5748  030370  000004                  .WORD   L10130-.
5749  030372  000000          5$:     0                       ;THIS LOCATION IS A FLAG, IT STARTS AT 0,
5750                                                          ;AND IS SET TO -1 WHEN LAST MEMORY ADDRESS
5751                                                          ;IS USED, TEST IS THEN ENDED WHEN PATTERN IS FINISHED
5752  030374                  ENDTST
5753  030374                  L10130:
5754  030374  104401                  TRAP    C$ETST
5755                          ;$MEM1
5756                          ;$MEM0
5757                          ;$MEM2  1K
5758                          ;$MEM3  1K
5759
5760  030376                          BADHEAD
5761                                  ;********************* TEST 48 *********************
5762                                  ;*ALU C BIT TEST
5763                                  ;*TEST THAT AN ADD OF 377 AND 377 WILL SET THE C BIT
5764  030376                          BADHEAD
5765                                  ;********************* TEST 48 *********************
5766
5767  030376                  BGNTST
5768  030376                  T48::
5769  030376                          MYINT
5770  030402                          MSTCLR                          ;MASTER CLEAR M8200,4,6,7
5771  030406  004737  003672          JSR     PC,MEMLD                ;LOAD MAINMEM DATA
5772  030412  030532                  TDATA                           ;POINTER TO DATA
5773  030414  004737  004044          JSR     PC,SPLD                 ;LOAD SP DATA
5774  030420  030532                  TDATA                           ;POINTER TO DATA
5775  030422                          BGNSEG
5776  030422  104404                  TRAP    C$BSEG
5777  030424                  1$:
5778  030424                          ROMCLK                          ;NEXT WORD IS INSTRUCTION,  BBN
```

```
5779  030430  010000                          010000                  ;MAR 0
5780  030432                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5781  030436  054400                          054400!<0*20>           ;ADD 377 AND 377, TO SET C BIT
5782  030440                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5783  030444  040421                          040401!<1*20>           ;ADD 0 AND 0 AND THE C BIT
5784  030446                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,  BBN
5785  030452  061224                          61224                   ;PUT RESULTS IN PORT4
5786  030454  012737  000001  002612          MOV     #1,$GDDAT               ;PUT "EXPECTED" IN $GDDAT
5787  030462  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
5788  030466  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
5789  030472  001413                          BEQ     2$              ;BR IF YES
5790  030474                                  ERROR   34              ;ERROR C BIT NOT SET
5791  030512  104455                          TRAP    C$ERDF
5792  030514  000042                          .WORD   34
5793  030516  005704                          .WORD   EM34
5794  030520  012132                          .WORD   ERR34
5795  030522                          2$:     ESCAPE  SEG
5796  030522  104410                          TRAP    C$ESCAPE
5797  030524  000002                          .WORD   10000$-.
5798  030526                                  ENDSEG
5799  030526                          10000$:
5800  030526  104405                          TRAP    C$ESEG
5801  030530                          ENDTST
5802  030530                          L10131:
5803  030530  104401                          TRAP    C$ETST
5804  030532    377     000     000   TDATA:  .BYTE   -1,0,0,0,0,0,0,0
5805  030535    000     000     000
5806  030540    000     000
5807
5808                                  .EVEN
5809
5810  030542                                  BADHEAD
5811                                          ;******************** TEST 49 **********************
5812                                          ;*ALU TEST
5813                                          ;*TEST OF ALU FUNCTION SEL B WITH C BIT CLEARED
5814                                          ;*ALU FUNCTION (B)    CODE=11
5815                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
5816                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
5817  030542                                  BADHEAD
5818                                          ;******************** TEST 49 **********************
5819
5820  030542                          BGNTST
5821  030542                          T49::
5822  030542                                  MYINT
5823  030546
5824  030552  005005                          MSTCLR                  ;MASTER CLEAR M8200,4,6,7
5825  030554  012702  030740                  CLR     R5              ;MEM + SP ADDRESS
5826  030560  004737  003672                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
5827  030564  002642                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
5828  030566  004737  004044                  MEMDAT                  ;POINTER TO DATA
5829  030572  002652                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
5830  030574                                  SPDAT                   ;POINTER TO DATA
5831  030574  104404                          BGNSEG
5832  030576  004737  004112          1$:     TRAP    C$BSEG
5833  030602  042737  000017  030620          JSR     PC,CLRC         ;CLEAR C BIT!
5834  030610  050537  030620                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
                                              BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
```

```
5835  030614
5836  030620  010000                    2$:     ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5837  030622  042737  000017  030640            010000                  ;LOAD MAR
5838  030630  050537  030640                    BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
5839  030634                                    BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
5840  030640  040620                    3$:     ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5841  030642                                    040400!<11*20>          ;BR   SEL B
5842  030646  061224                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5843  030650  111237  002612                    61224                   ;MOVE BR TO PORT4
5844  030654  116104  000004                    MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
5845  030660  123704  002612                    MOVB    4(R1),R4        ;PUT "FOUND" IN R4
5846  030664  001413                            CMPB    $GDDAT,R4               ;DATA CORRECT?
5847  030666                                    BEQ     4$              ;BR IF YES
5848  030704  104455                            ERROR   15              ;ALU ERROR
5849  030706  000017                            TRAP    C$ERDF
5850  030710  005223                            .WORD   15
5851  030712  010114                            .WORD   EM15
5852  030714                                    .WORD   ERR15
5853  030714  104410                    4$:     ESCAPE  SEG
5854  030716  000014                            TRAP    C$ESCAPE
5855  030720  005202                            .WORD   10000$-.
5856  030722  005205                            INC     R2              ;NEXT DATA
5857  030724  022705  000010                    INC     R5              ;NEXT ADDRESS
5858  030730  001322                            CMP     #10,R5          ;DONE YET?
5859  030732                                    BNE     1$              ;BR IF NO
5860  030732                                    ENDSEG
5861  030732  104405                    10000$:
5862  030734                                    TRAP    C$ESEG
5863  030734  104432                            EXIT    TST
5864  030736  000012                            TRAP    C$EXIT
5865  030740     000     377     000    5$:     .WORD   L10132-.
5866  030743     377     125     252            .BYTE   0,-1,0,-1,125,252,125,252
5867  030746     125     252
5868
5869                                    .EVEN
5870  030750                            ENDTST
5871  030750                            L10132:
5872  030750  104401                            TRAP    C$ETST
5873
5874  030752                                    BADHEAD
5875                                            ;********************* TEST 50 *********************
5876                                            ;*ALU TEST
5877                                            ;*TEST OF ALU FUNCTION SEL A WITH C BIT CLEARED
5878                                            ;*ALU FUNCTION (A)     CODE=10
5879                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
5880                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
5881  030752                                    BADHEAD
5882                                            ;********************* TEST 50 *********************
5883
5884  030752                            BGNTST
5885  030752                            T50::
5886  030752                                    MYINT
5887  030756                                    MSTCLR                  ;MASTER CLEAR M8200,4,6,7
5888  030762  005005                            CLR     R5              ;MEM + SP ADDRESS
5889  030764  012702  031150                    MOV     #5$,R2          ;POINTER TO CORRECT DATA
5890  030770  004737  003672                    JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
```

```
5891  030774  002642                        MEMDAT                  ;POINTER TO DATA
5892  030776  004737  004044                JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
5893  031002  002652                        SPDAT                   ;POINTER TO DATA
5894  031004                                BGNSEG
5895  031004  104404                        TRAP    C$BSEG
5896  031006  004737  004112        1$:     JSR     PC,CLRC         ;CLEAR C BIT!
5897  031012  042737  000017  031030        BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
5898  031020  050537  031030                BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
5899  031024                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5900  031030  010000                2$:     010000                  ;LOAD MAR
5901  031032  042737  000017  031050        BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
5902  031040  050537  031050                BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
5903  031044                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5904  031050  040600                3$:     040400!<10*20>          ;BR    SEL A
5905  031052                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5906  031056  061224                        61224                   ;MOVE BR TO PORT4
5907  031060  111237  002612                MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
5908  031064  116104  000004                MOVB    4(R1),R4        ;PUT "FOUND" IN R4
5909  031070  123704  002612                CMPB    $GDDAT,R4               ;DATA CORRECT?
5910  031074  001413                        BEQ     4$              ;BR IF YES
5911  031076                                ERROR   15              ;ALU ERROR
5912  031114  104455                        TRAP    C$ERDF
5913  031116  000017                        .WORD   15
5914  031120  005223                        .WORD   EM15
5915  031122  010114                        .WORD   ERR15
5916  031124                        4$:     ESCAPE  SEG
5917  031124  104410                        TRAP    C$ESCAPE
5918  031126  000014                        .WORD   10000$-.
5919  031130  005202                        INC     R2              ;NEXT DATA
5920  031132  005205                        INC     R5              ;NEXT DATA
5921  031134  022705  000010                CMP     #10,R5          ;DONE YET?
5922  031140  001322                        BNE     1$              ;BR IF NO
5923  031142                                ENDSEG
5924  031142                        10000$:
5925  031142  104405                        TRAP    C$ESEG
5926  031144                                EXIT    TST
5927  031144  104432                        TRAP    C$EXIT
5928  031146  000012                        .WORD   L10133-.
5929  031150     000     000     377  5$:     .BYTE   0,0,-1,-1,125,125,252,252
5930  031153     377     125     125
5931  031156     252     252
5932
5933                                .EVEN
5934  031160                        ENDTST
5935  031160                        L10133:
5936  031160  104401                        TRAP    C$ETST
5937
5938  031162                                BADHEAD
5939                                        ;********** TEST 51 **********
5940                                        ;*ALU TEST
5941                                        ;*TEST OF ALU FUNCTION A OR NOTB WITH C BIT CLEARED
5942                                        ;*ALU FUNCTION (A OR NOTB)    CODE=12
5943                                        ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
5944                                        ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
5945  031162                                BADHEAD
5946                                        ;********** TEST 51 **********
```

```
031162                                   BGNTST
031162                                   T51::
031162                                           MYINT
031166                                           MSTCLR                  ;MASTER CLEAR M8200,4,6,7
031172  005005                                   CLR     R5              ;MEM + SP ADDRESS
031174  012702  031360                           MOV     #5$,R2          ;POINTER TO CORRECT DATA
031200  004737  003672                           JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
031204  002642                                   MEMDAT                  ;POINTER TO DATA
031206  004737  004044                           JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
031212  002652                                   SPDAT                   ;POINTER TO DATA
031214                                           BGNSEG
031214  104404                                   TRAP    C$BSEG
031216  004737  004112           1$:             JSR     PC,CLRC         ;CLEAR C BIT!
031222  042737  000017  031240                   BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
031230  050537  031240                           BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
031234                                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031240  010000                   2$:             010000                  ;LOAD MAR
031242  042737  000017  031260                   BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
031250  050537  031260                           BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
031254                                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031260  040640                   3$:             040400!<12*20>          ;BR   A OR NOTB
031262                                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031266  061224                                   61224                   ;MOVE BR TO PORT4
031270  111237  002612                           MOVB    (R2),$GDDAT             ;PUT ''EXPECTED'' IN $GDDAT
031274  116104  000004                           MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
031300  123704  002612                           CMPB    $GDDAT,R4               ;DATA CORRECT?
031304  001413                                   BEQ     4$              ;BR IF YES
031306                                           ERROR   15              ;ALU ERROR
031324  104455                                   TRAP    C$ERDF
031326  000017                                   .WORD   15
031330  005223                                   .WORD   EM15
031332  010114                                   .WORD   ERR15
031334                           4$:             ESCAPE  SEG
031334  104410                                   TRAP    C$ESCAPE
031336  000014                                   .WORD   10000$-.
031340  005202                                   INC     R2              ;NEXT DATA
031342  005205                                   INC     R5              ;NEXT DATA
031344  022705  000010                           CMP     #10,R5          ;DONE YET?
031350  001322                                   BNE     1$              ;BR IF NO
031352                                           ENDSEG
031352                           10000$:
031352  104405                                   TRAP    C$ESEG
031354                                           EXIT    TST
031354  104432                                   TRAP    C$EXIT
031356  000012                                   .WORD   L10134-.
031360     377     000     377   5$:             .BYTE   -1,0,-1,-1,-1,125,252,-1
031363     377     377     125
031366     252     377

                                 .EVEN
031370                           ENDTST
031370                           L10134:
031370  104401                                   TRAP    C$ETST

031372                                           BADHEAD
```

```
                                              ;******************** TEST 52 *********************
                                              ;*ALU TEST
                                              ;*TEST OF ALU FUNCTION A AND B WITH C BIT CLEARED
                                              ;*ALU FUNCTION (A AND B)     CODE=13
                                              ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                              ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
031372                                        BADHEAD
                                              ;******************** TEST 52 *********************

031372                              BGNTST
031372                              T52::
031372                                        MYINT
031376                                        MSTCLR                       ;MASTER CLEAR M8200,4,6,7
031402  005005                                CLR     R5                   ;MEM + SP ADDRESS
031404  012702  031570                        MOV     #5$,R2               ;POINTER TO CORRECT DATA
031410  004737  003672                        JSR     PC,MEMLD             ;LOAD 8 WORDS OF MAIN MEMORY
031414  002642                                MEMDAT                       ;POINTER TO DATA
031416  004737  004044                        JSR     PC,SPLD              ;LOAD 8 WORDS OF SP
031422  002652                                SPDAT                        ;POINTER TO DATA
031424                                        BGNSEG
031424  104404                                TRAP    C$BSEG
031426  004737  004112          1$:           JSR     PC,CLRC              ;CLEAR C BIT!
031432  042737  000017  031450                BIC     #17,2$               ;CLEAR ADDRESS FIELD OF INSTRUCTION
031440  050537  031450                        BIS     R5,2$                ;ADD ADDRESS TO INSTRUCTION
031444                                        ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031450  010000                  2$:           010000                       ;LOAD MAR
031452  042737  000017  031470                BIC     #17,3$               ;CLEAR ADDRESS OF INSTRUCTION
031460  050537  031470                        BIS     R5,3$                ;ADD ADDRESS TO INSTRUCTION
031464                                        ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031470  040660                  3$:           040400!<13*20>               ;BR    A AND B
031472                                        ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031476  061224                                61224                        ;MOVE BR TO PORT4
031500  111237  002612                        MOVB    (R2),$GDDAT              ;PUT ''EXPECTED'' IN $GDDAT
031504  116104  000004                        MOVB    4(R1),R4             ;PUT ''FOUND'' IN R4
031510  123704  002612                        CMPB    $GDDAT,R4                ;DATA CORRECT?
031514  001413                                BEQ     4$                   ;BR IF YES
031516                                        ERROR   15                   ;ALU ERROR
031534  104455                                TRAP    C$ERDF
031536  000017                                .WORD   15
031540  005223                                .WORD   EM15
031542  010114                                .WORD   ERR15
031544                          4$:           ESCAPE  SEG
031544  104410                                TRAP    C$ESCAPE
031546  000014                                .WORD   10000$-.
031550  005202                                INC     R2                   ;NEXT DATA
031552  005205                                INC     R5                   ;NEXT DATA
031554  022705  000010                        CMP     #10,R5               ;DONE YET?
031560  001322                                BNE     1$                   ;BR IF NO
031562                                        ENDSEG
031562                          10000$:
031562  104405                                TRAP    C$ESEG
031564                                        EXIT    TST
031564  104432                                TRAP    C$EXIT
031566  000012                                .WORD   L10135-.
031570     000     000     000  5$:           .BYTE   0,0,0,-1,125,0,0,252
031573     377     125     000
```

```
031576     000      252

                                  .EVEN
031600                            ENDTST
031600                            L10135:
031600  104401                            TRAP    C$ETST

031602                                    BADHEAD
                                          ;******************** TEST 53 ********************
                                          ;*ALU TEST
                                          ;*TEST OF ALU FUNCTION A OR B WITH C BIT CLEARED
                                          ;*ALU FUNCTION (A OR B)    CODE=14
                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
031602                                    BADHEAD

                                          ;******************** TEST 53 ********************

031602                            BGNTST
031602                            T53::
031602                                    MYINT
031606                                    MSTCLR                  ;MASTER CLEAR M8200,4,6,7
031612  005005                            CLR     R5              ;MEM + SP ADDRESS
031614  012702  032000                    MOV     #5$,R2          ;POINTER TO CORRECT DATA
031620  004737  003672                    JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
031624  002642                            MEMDAT                  ;POINTER TO DATA
031626  004737  004044                    JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
031632  002652                            SPDAT                   ;POINTER TO DATA
031634                                    BGNSEG
031634  104404                            TRAP    C$BSEG
031636  004737  004112            1$:     JSR     PC,CLRC         ;CLEAR C BIT!
031642  042737  000017  031660            BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
031650  050537  031660                    BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
031654                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031660  010000                    2$:     010000                  ;LOAD MAR
031662  042737  000017  031700            BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
031670  050537  031700                    BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
031674                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031700  040700                    3$:     040400!<14*20>          ;BR    A OR B
031702                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031706  061224                            61224                   ;MOVE BR TO PORT4
031710  111237  002612                    MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
031714  116104  000004                    MOVB    4(R1),R4        ;PUT "FOUND" IN R4
031720  123704  002612                    CMPB    $GDDAT,R4               ;DATA CORRECT?
031724  001413                            BEQ     4$              ;BR IF YES
031726                                    ERROR   15              ;ALU ERROR
031744  104455                            TRAP    C$ERDF
031746  000017                            .WORD   15
031750  005223                            .WORD   EM15
031752  010114                            .WORD   ERR15
031754                            4$:     ESCAPE  SEG
031754  104410                            TRAP    C$ESCAPE
031756  000014                            .WORD   10000$-.
031760  005202                            INC     R2              ;NEXT DATA
031762  005205                            INC     R5              ;NEXT DATA
031764  022705  000010                    CMP     #10,R5          ;DONE YET?
031770  001322                            BNE     1$              ;BR IF NO
```

```
 6115  031772                                  ENDSEG
 6116  031772                            10000$:
 6117  031772  104405                          TRAP    C$ESEG
 6118  031774                                  EXIT    TST
 6119  031774  104432                          TRAP    C$EXIT
 6120  031776  000012                          .WORD   L10136-.
 6121  032000     000     377     377  5$:     .BYTE   0,-1,-1,-1,125,-1,-1,252
 6122  032003     377     125     377
 6123  032006     377     252
 6124
 6125                                  .EVEN
 6126  032010                          ENDTST
 6127  032010                          L10136:
 6128  032010  104401                          TRAP    C$ETST
 6129
 6130  032012                                  BADHEAD
 6131                                          ;******************** TEST 54 ********************
 6132                                          ;*ALU TEST
 6133                                          ;*TEST OF ALU FUNCTION A XOR B WITH C BIT CLEARED
 6134                                          ;*ALU FUNCTION (A XOR B)     CODE=15
 6135                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 6136                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 6137  032012                                  BADHEAD
 6138                                          ;******************** TEST 54 ********************
 6139
 6140  032012                          BGNTST
 6141  032012                          T54::
 6142  032012                                  MYINT
 6143  032016                                  MSTCLR                  ;MASTER CLEAR M8200,4,6,7
 6144  032022  005005                          CLR     R5              ;MEM + SP ADDRESS
 6145  032024  012702  032210                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
 6146  032030  004737  003672                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 6147  032034  002642                          MEMDAT                  ;POINTER TO DATA
 6148  032036  004737  004044                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 6149  032042  002652                          SPDAT                   ;POINTER TO DATA
 6150  032044                                  BGNSEG
 6151  032044  104404                          TRAP    C$BSEG
 6152  032046  004737  004112          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
 6153  032052  042737  000017  032070          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 6154  032060  050537  032070                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 6155  032064                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 6156  032070  010000                  2$:     010000                  ;LOAD MAR
 6157  032072  042737  000017  032110          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 6158  032100  050537  032110                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 6159  032104                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 6160  032110  040720                  3$:     040400!<15*20>          ;BR    A XOR B
 6161  032112                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 6162  032116  061224                          61224                   ;MOVE BR TO PORT4
 6163  032120  111237  002612                  MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
 6164  032124  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
 6165  032130  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
 6166  032134  001413                          BEQ     4$              ;BR IF YES
 6167  032136                                  ERROR   15              ;ALU ERROR
 6168  032154  104455                          TRAP    C$ERDF
 6169  032156  000017                          .WORD   15
 6170  032160  005223                          .WORD   EM15
```

```
6171  032162  010114                          .WORD   ERR15
6172  032164                          4$:     ESCAPE  SEG
6173  032164  104410                          TRAP    C$ESCAPE
6174  032166  000014                          .WORD   10000$-.
6175  032170  005202                          INC     R2              ;NEXT DATA
6176  032172  005205                          INC     R5              ;NEXT DATA
6177  032174  022705  000010                  CMP     #10,R5          ;DONE YET?
6178  032200  001322                          BNE     1$              ;BR IF NO
6179  032202                                  ENDSEG
6180  032202                          10000$:
6181  032202  104405                          TRAP    C$ESEG
6182  032204                                  EXIT    TST
6183  032204  104432                          TRAP    C$EXIT
6184  032206  000012                          .WORD   L10137-.
6185  032210     000     377     377  5$:     .BYTE   0,-1,-1,0,0,-1,-1,0
6186  032213     000     000     377
6187  032216     377     000
6188
6189                                  .EVEN
6190  032220                          ENDTST
6191  032220                          L10137:
6192  032220  104401                          TRAP    C$ETST
6193
6194  032222                                  BADHEAD
6195                                  ;********************* TEST 55 *********************
6196                                  ;*ALU TEST
6197                                  ;*TEST OF ALU FUNCTION ADD WITH C BIT CLEARED
6198                                  ;*ALU FUNCTION (A PLUS B)     CODE=00
6199                                  ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6200                                  ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
6201  032222                          BADHEAD
6202                                  ;********************* TEST 55 *********************
6203
6204  032222                          BGNTST
6205  032222                          T55::
6206  032222                                  MYINT
6207  032226                                  MSTCLR                  ;MASTER CLEAR M8200,4,6,7
6208  032232  005005                          CLR     R5              ;MEM + SP ADDRESS
6209  032234  012702  032420                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
6210  032240  004737  003672                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
6211  032244  002642                          MEMDAT                  ;POINTER TO DATA
6212  032246  004737  004044                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
6213  032252  002652                          SPDAT                   ;POINTER TO DATA
6214  032254                                  BGNSEG
6215  032254  104404                          TRAP    C$BSEG
6216  032256  004737  004112          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
6217  032262  042737  000017  032300          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
6218  032270  050537  032300                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
6219  032274                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6220  032300  010000                  2$:     010000                  ;LOAD MAR
6221  032302  042737  000017  032320          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
6222  032310  050537  032320                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
6223  032314                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6224  032320  040400                  3$:     040400!<00*20>          ;BR   ADD
6225  032322                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6226  032326  061224                          61224                   ;MOVE BR TO PORT4
```

```
6227  032330  111237  002612                  MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
6228  032334  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
6229  032340  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
6230  032344  001413                          BEQ     4$              ;BR IF YES
6231  032346                                  ERROR   15              ;ALU ERROR
6232  032364  104455                          TRAP    C$ERDF
6233  032366  000017                          .WORD   15
6234  032370  005223                          .WORD   EM15
6235  032372  010114                          .WORD   ERR15
6236  032374                          4$:     ESCAPE  SEG
6237  032374  104410                          TRAP    C$ESCAPE
6238  032376  000014                          .WORD   10000$-.
6239  032400  005202                          INC     R2              ;NEXT DATA
6240  032402  005205                          INC     R5              ;NEXT DATA
6241  032404  022705  000010                  CMP     #10,R5          ;DONE YET?
6242  032410  001322                          BNE     1$              ;BR IF NO
6243  032412                                  ENDSEG
6244  032412                          10000$:
6245  032412  104405                          TRAP    C$ESEG
6246  032414                                  EXIT    TST
6247  032414  104432                          TRAP    C$EXIT
6248  032416  000012                          .WORD   L10140-.
6249  032420     000     377     377  5$:     .BYTE   0,-1,-1,376,252,-1,-1,124
6250  032423     376     252     377
6251  032426     377     124
6252
6253                                  .EVEN
6254  032430                          ENDTST
6255  032430                          L10140:
6256  032430  104401                          TRAP    C$ETST
6257
6258  032432                                  BADHEAD
6259                                          ;******************** TEST 56 *********************
6260                                          ;*ALU TEST
6261                                          ;*TEST OF ALU FUNCTION 2A W.C WITH C BIT CLEARED
6262                                          ;*ALU FUNCTION (A PLUS A PLUS C)         CODE=6
6263                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6264                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
6265  032432                                  BADHEAD
6266                                          ;******************** TEST 56 *********************
6267
6268  032432                          BGNTST
6269  032432                          T56::
6270  032432                                  MYINT
6271  032436                                  MSTCLR                          ;MASTER CLEAR M8200,4,6,7
6272  032442  005005                          CLR     R5                      ;MEM + SP ADDRESS
6273  032444  012702  032630                  MOV     #5$,R2                  ;POINTER TO CORRECT DATA
6274  032450  004737  003672                  JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
6275  032454  002642                          MEMDAT                          ;POINTER TO DATA
6276  032456  004737  004044                  JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
6277  032462  002652                          SPDAT                           ;POINTER TO DATA
6278  032464                                  BGNSEG
6279  032464  104404                          TRAP    C$BSEG
6280  032466  004737  004112          1$:     JSR     PC,CLRC                 ;CLEAR C BIT!
6281  032472  042737  000017  032510          BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
6282  032500  050537  032510                  BIS     R5,2$                   ;ADD ADDRESS TO INSTRUCTION
```

```
032504                                           ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
032510  010000                           2$:     010000                   ;LOAD MAR
032512  042737  000017  032530                   BIC     #17,3$           ;CLEAR ADDRESS OF INSTRUCTION
032520  050537  032530                           BIS     R5,3$            ;ADD ADDRESS TO INSTRUCTION
032524                                           ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
032530  040540                           3$:     040400!<6*20>            ;BR    2A W/C
032532                                           ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
032536  061224                                   61224                    ;MOVE BSR TO PORT4
032540  111237  002612                           MOVB    (R2),$GDDAT              ;PUT ''EXPECTED'' IN $GDDAT
032544  116104  000004                           MOVB    4(R1),R4         ;PUT ''FOUND'' IN R4
032550  123704  002612                           CMPB    $GDDAT,R4                ;DATA CORRECT?
032554  001413                                   BEQ     4$               ;BR IF YES
032556                                           ERROR   15               ;ALU ERROR
032574  104455                                   TRAP    C$ERDF
032576  000017                                   .WORD   15
032600  005223                                   .WORD   EM15
032602  010114                                   .WORD   ERR15
032604                                   4$:     ESCAPE  SEG
032604  104410                                   TRAP    C$ESCAPE
032606  000014                                   .WORD   10000$-.
032610  005202                                   INC     R2               ;NEXT DATA
032612  005205                                   INC     R5               ;NEXT ADDRESS
032614  022705  000010                           CMP     #10,R5           ;DONE YET?
032620  001322                                   BNE     1$               ;BR IF NO
032622                                           ENDSEG
032622                                   10000$:
032622  104405                                   TRAP    C$ESEG
032624                                           EXIT    TST
032624  104432                                   TRAP    C$EXIT
032626  000012                                   .WORD   L10141-.
032630     000     000     376           5$:     .BYTE   0,0,376,376,252,252,124,124
032633     376     252     252
032636     124     124


                                         .EVEN
032640                                   ENDTST
032640                                   L10141:
032640  104401                                   TRAP    C$ETST

032642                                           BADHEAD
                                                 ;******************** TEST 57 ********************
                                                 ;*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
032642                                           BADHEAD
                                                 ;******************** TEST 57 ********************

032642                                   BGNTST
032642                                   T57::
032642                                           MYINT
032646                                           MSTCLR                   ;MASTER CLEAR M8200,4,6,7
032652  005005                                   CLR     R5               ;MEM + SP ADDRESS
032654  012702  033042                           MOV     #5$,R2           ;POINTER TO CORRECT DATA
032660  004737  003672                           JSR     PC,MEMLD         ;LOAD 8 WORDS OF MAIN MEMORY
```

```
6339  032664  002642                      MEMDAT                  ;POINTER TO DATA
6340  032666  004737  004044              JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
6341  032672  002652                      SPDAT                   ;POINTER TO DATA
6342  032674                              BGNSEG
6343  032674  104404                      TRAP    C$BSEG
6344  032676  004737  004112      1$:     JSR     PC,CLRC         ;CLEAR C BIT!
6345  032702  042737  000017  032720      BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
6346  032710  050537  032720              BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
6347  032714                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6348  032720  010000              2$:     010000                  ;LOAD MAR
6349  032722  042737  000017  032740      BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
6350  032730  050537  032740              BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
6351  032734                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6352  032740  040740              3$:     040400!<16*20>          ;BR _ SUB
6353  032742                              ROMCLK
6354  032746  061224                      61224                   ;MOVE BR TO PORT4
6355  032750  111237  002612              MOVB    (R2),$GDDAT             ;PUT ''EXPECTED'' IN $GDDAT
6356  032754  116104  000004              MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
6357  032760  123737  002612  002612      CMPB    $GDDAT,$GDDAT           ;DATA CORRECT?
6358  032766  001413                      BEQ     4$              ;BR IF YES
6359  032770                              ERROR   15              ;ALU ERROR
6360  033006  104455                      TRAP    C$ERDF
6361  033010  000017                      .WORD   15
6362  033012  005223                      .WORD   EM15
6363  033014  010114                      .WORD   ERR15
6364  033016                      4$:     ESCAPE  SEG
6365  033016  104410                      TRAP    C$ESCAPE
6366  033020  000014                      .WORD   10000$-.
6367  033022  005202                      INC     R2              ;NEXT DATA
6368  033024  005205                      INC     R5              ;NEXT ADDRESS
6369  033026  022705  000010              CMP     #10,R5          ;DONE YET?
6370  033032  001321                      BNE     1$              ;BR IF NO
6371  033034                              ENDSEG
6372  033034                      10000$:
6373  033034  104405                      TRAP    C$ESEG
6374  033036                              EXIT    TST
6375  033036  104432                      TRAP    C$EXIT
6376  033040  000012                      .WORD   L10142-.
6377  033042     000     001     377  5$: .BYTE   0,1,-1,0,0,253,125,0
6378  033045     000     000     253
6379  033050     125     000
6380
6381
6382                              .EVEN
6383  033052                      ENDTST
6384  033052                      L10142:
6385  033052  104401                      TRAP    C$ETST
6386
6387
6388  033054                              BADHEAD
6389                                      ;********************* TEST 58 **********************
6390                                      ;*ALU TEST
6391                                      ;*TEST OF ALU FUNCTION ADD W/C WITH C BIT CLEARED
6392                                      ;*ALU FUNCTION (A PLUS B PLUS C)          CODE=01
6393                                      ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6394                                      ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
```

```
                                        BADHEAD
                                        ;******************** TEST 58 ********************

                                BGNTST
                                T58::
                                        MYINT
                                        MSTCLR                          ;MASTER CLEAR M8200,4,6,7
                                        CLR     R5                      ;MEM + SP ADDRESS
                                        MOV     #5$,R2                  ;POINTER TO CORRECT DATA
                                        JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
                                        MEMDAT                          ;POINTER TO DATA
                                        JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
                                        SPDAT                           ;POINTER TO DATA
                                        BGNSEG
                                        TRAP    C$BSEG
                                1$:     JSR     PC,CLRC                 ;CLEAR C BIT!
                                        BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
                                        BIS     R5,2$                   ;ADD ADDRESS TO INSTRUCTION
                                        ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
                                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
                                3$:     040400!<01*20>                  ;BR   ADD W/C
                                        ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
                                        61224                           ;MOVE BR TO PORT4
                                        MOVB    (R2),$GDDAT                     ;PUT "EXPECTED" IN $GDDAT
                                        MOVB    4(R1),R4                ;PUT "FOUND" IN R4
                                        CMPB    $GDDAT,R4                       ;DATA CORRECT?
                                        BEQ     4$                      ;BR IS YES
                                        ERROR   15                      ;ALU ERROR
                                        TRAP    C$ERDF
                                        .WORD   15
                                        .WORD   EM15
                                        .WORD   ERR15
                                4$:     ESCAPE  SEG
                                        TRAP    C$ESCAPE
                                        .WORD   10000$-.
                                        INC     R2                      ;NEXT DATA
                                        INC     R5                      ;NEXT ADDRESS
                                        CMP     #10,R5                  ;DONE YET?
                                        BNE     1$                      ;BR IF NO
                                        ENDSEG
                                10000$:
                                        TRAP    C$ESEG
                                        EXIT    TST
                                        TRAP    C$EXIT
                                        .WORD   L10143-.
                                5$:     .BYTE   0,-1,-1,376,252,-1,-1,124

                                        .EVEN
                                ENDTST
                                L10143:
                                        TRAP    C$ETST
```

```
033264                      BADHEAD
                            ;******************** TEST 59 ********************
                            ;*ALU TEST
                            ;*TEST OF ALU FUNCTION SUB W/C WITH C BIT CLEARED
                            ;*ALU FUNCTION (A-B-C)  CODE=2
                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
033264                      BADHEAD
                            ;******************** TEST 59 ********************

033264              BGNTST
033264              T59::
033264                      MYINT
```

```
6466  033270
6467  033274  005005                    MSTCLR
6468  033276  012702  033462            CLR     R5              ;MASTER CLEAR M8200,4,6,7
6469  033302  004737  003672            MOV     #5$,R2          ;MEM + SP ADDRESS
6470  033306  002642                    JSR     PC, MEMLD       ;POINTER TO CORRECT DATA
6471  033310  004737  004044            MEMDAT                  ;LOAD 8 WORDS OF MAIN MEMORY
6472  033314  002652                    JSR     PC,SPLD         ;POINTER TO DATA
6473  033316                            SPDAT                   ;LOAD 8 WORDS OF SP
6474  033316  104404                    BGNSEG                  ;POINTER TO DATA
6475  033320  004737  004112            TRAP    C$BSEG
6476  033324  042737  000017  033342  1$:     JSR     PC,CLRC         ;CLEAR C BIT!
6477  033332  050537  033342            BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
6478  033336                            BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
6479  033342  010000                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6480  033344  042737  000017  033362  2$:     010000          ;LOAD MAR
6481  033352  050537  033362            BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
6482  033356                            BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
6483  033362  040440                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6484  033364                    3$:     040400!<2*20>           ;BR _ SUB W/C
6485  033370  061224                    ROMCLK
6486  033372  111237  002612            61224           ;MOVE BR TO PORT4
6487  033376  116104  000004            MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
6488  033402  123704  002612            MOVB    4(R1),R4        ;PUT"FOUND" IN R4
6489  033406  001413                    CMPB    $GDDAT,R4               ;DATA CORRECT?
6490  033410                            BEQ     4$              ;BR IF YES
6491  033426  104455                    ERROR   15              ;ALU ERROR
6492  033430  000017                    TRAP    C$ERDF
6493  033432  005223                    .WORD   15
6494  033434  010114                    .WORD   EM15
6495  033436                            .WORD   ERR15
6496  033436  104410            4$:     ESCAPE  SEG
6497  033440  000014                    TRAP    C$ESCAPE
6498  033442  005202                    .WORD   10000$-.
6499  033444  005205                    INC     R2                      ;NEXT DATA
6500  033446  022705  000010            INC     R5                      ;NEXT ADDRESS
6501  033452  001322                    CMP     #10,R5                  ;DONE YET?
6502  033454                            BNE     1$                      ;BR IF NO
6503  033454                            ENDSEG
6504  033454  104405            10000$:
6505  033456                            TRAP    C$ESEG
6506  033456  104432                    EXIT    TST
6507  033460  000012                    TRAP    C$EXIT
6508  033462     377     000     376    .WORD   L10144-.
6509  033465     377     377     252    5$:     .BYTE   -1,0,376,-1,-1,252,124,-1
6510  033470     124     377
6511
6512
6513                                    .EVEN
6514  033472                            ENDTST
6515  033472                            L10144:
6516  033472  104401                    TRAP    C$ETST
6517
6518
6519  033474                            BADHEAD
6520                                    ;********************* TEST 60 *********************
6521                                    ;*ALU TEST
```

```
                                        ;*TEST OF ALU FUNCTION INC A WITH C BIT CLEARED
                                        ;*ALU FUNCTION (A PLUS 1)       CODE=3
                                        ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                        ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
033474                                  BADHEAD
                                        ;********************* TEST 60 **********************

033474                          BGNTST
033474                          T60::
033474                                  MYINT
033500                                  MSTCLR                  ;MASTER CLEAR M8200,4,6,7
033504  012702  033672                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
033510  005005                          CLR     R5
033512  004737  003672                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMRY
033516  002642                          MEMDAT                  ;POINTER TO DATA
033520  004737  004044                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
033524  002652                          SPDAT                   ;POINTER TO DATA
033526                                  BGNSEG
033526  104404                          TRAP    C$BSEG
033530  004737  004112          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
033534  042737  000017  033552          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
033542  050537  033552                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
033546                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033552  010000                  2$:     010000                  ;LOAD MAR
033554  042737  000017  033572          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
033562  050537  033572                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
033566                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033572  040460                  3$:     040400!<3*20>           ;BR  INC A
033574                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033600  061224                          61224                   ;MOVE BR TO PORT4
033602  111237  002612                  MOVB    (R2),$GDDAT             ;PUT 'EXPECTED' IN $GDDAT
033606  116104  000004                  MOVB    4(R1),R4        ;PUT 'FOUND' IN R4
033612  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
033616  001413                          BEQ     4$              ;BR IF YES
033620                                  ERROR   15              ;ALU ERROR
033636  104455                          TRAP    C$ERDF
033640  000017                          .WORD   15
033642  005223                          .WORD   EM15
033644  010114                          .WORD   ERR15
033646                          4$:     ESCAPE  SEG
033646  104410                          TRAP    C$ESCAPE
033650  000014                          .WORD   10000$-.
033652  005202                          INC     R2              ;NEXT DATA
033654  005205                          INC     R5
033656  022705  000010                  CMP     #10,R5          ;DONE YET?
033662  001322                          BNE     1$              ;BR IF NO
033664                                  ENDSEG
033664                          10000$:
033664  104405                          TRAP    C$ESEG
033666                                  EXIT    TST
033666  104432                          TRAP    C$EXIT
033670  000012                          .WORD   L10145-.
033672    001     001     000   5$:     .BYTE   1,1,0,0,126,126,253,253
033675    000     126     126
033700    253     253
```

```
                                .EVEN
                                ENDTST
                                L10145:
                                        TRAP    C$ETST


                                        BADHEAD
                                        ;********************* TEST 61 *********************
                                        ;*ALU TEST
                                        ;*TEST OF ALU FUNCTION 2A WITH C BIT CLEARED
                                        ;*ALU FUNCTION (A PLUS A)       CODE=5
                                        ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                        ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
                                        BADHEAD
                                        ;********************* TEST 61 *********************

                                BGNTST
                                T61::
                                        MYINT
                                        MSTCLR                          ;MASTER CLEAR DMC11
                                        CLR     R5                      ;MEM * SP ADDRESS
                                        MOV     #5$,R2                  ;POINTER TO CORRECT DATA
                                        JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
                                        MEMDAT                          ;POINTER TO DATA
                                        JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
                                        SPDAT                           ;POINTER TO DATA
                                        BGNSEG
                                        TRAP    C$BSEG
                                1$:     JSR     PC,CLRC                 ;CLEAR C BIT!
                                        BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
                                        BIS     R5,2$                   ;ADD ADDRESS TO INSTRUCTION
                                        ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
                                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
                                3$:     040400!<5*20>                   ;BR   2A
                                        ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
                                        61224                           ;MOVE BR TO PORT4
                                        MOVB    (R2),$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                      ;ALU ERROR
                                        TRAP    C$ERDF
                                        .WORD   15
                                        .WORD   EM15
                                        .WORD   ERR15
                                4$:     ESCAPE  SEG
                                        TRAP    C$ESCAPE
                                        .WORD   10000$-.
                                        INC     R2                      ;NEXT DATA
                                        INC     R5                      ;NEXT ADDRESS
                                        CMP     #10,R5                  ;DONE YET?
                                        BNE     1$                      ;BR IF NO
                                        ENDSEG
```

```
034074                                   10000$:
034074  104405                                    TRAP    C$ESEG
034076                                            EXIT    TST
034076  104432                                    TRAP    C$EXIT
034100  000012                                    .WORD   L10146-.
034102     000      000      376         5$:      .BYTE   0,0,376,376,252,252,124,124
034105     376      252      252
034110     124      124
                                         .EVEN
034112                                   ENDTST
034112                                   L10146:
034112  104401                                    TRAP    C$ETST
034114                                            BADHEAD
                                                  ;******************** TEST 62 ********************
                                                  ;*ALU TEST
                                                  ;*TEST OF ALU FUNCTION A PLUS C WITH C BIT CLEARED
                                                  ;*ALU FUNCTION (A PLUS C)        CODE=4
                                                  ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                                  ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
034114                                            BADHEAD
                                                  ;******************** TEST 62 ********************
034114                                   BGNTST
034114                                   T62::
034114                                            MYINT
034120                                            MSTCLR                          ;MASTER CLEAR M8200,4,6,7
034124  005005                                    CLR     R5                      ;MEM + SP ADDRESS
034126  012702  034312                            MOV     #5$,R2                  ;POINTER TO CORRECT DATA
034132  004737  003672                            JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
034136  002642                                    MEMDAT                          ;POINTER TO DATA
034140  004737  004044                            JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
034144  002652                                    SPDAT                           ;POINTER TO DATA
034146                                            BGNSEG
034146  104404                                    TRAP    C$BSEG
034150  004737  004112                   1$:      JSR     PC,CLRC                 ;CLEAR C BIT!
034154  042737  000017  034172                    BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
034162  050537  034172                            BIS     R5,2$                   ;ADD ADDRESS TO INSTRUCTION
034166                                            ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
034172  010000                           2$:      010000                          ;LOAD MAR
034174  042737  000017  034212                    BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
034202  050537  034212                            BIS     R5,3$                   ;ADD ADDRESS TO INSTRUCTION
034206                                            ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=304
034212  040500                           3$:      040400!<4*20>                   ;BR    A PLUS C
034214                                            ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCKL PC=5305
034220  061224                                    61224                           ;MOVE BR TO PORT4
034222  111237  002612                            MOVB    (R2),$GDDAT                     ;PUT "EXPECTED" IN $GDDAT
034226  116104  000004                            MOVB    4(R1),R4                ;PUT "FOUND" IN R4
034232  123704  002612                            CMPB    $GDDAT,R4                       ;DATA CORRECT?
034236  001413                                    BEQ     4$                      ;BR IS YES
034240                                            ERROR   15                      ;ALU ERROR
034256  104455                                    TRAP    C$ERDF
034260  000017                                    .WORD   15
034262  005223                                    .WORD   EM15
```

```
 6690  034264  010114                          .WORD   ERR15
 6691  034266                          4$:     ESCAPE  SEG
 6692  034266  104410                          TRAP    C$ESCAPE
 6693  034270  000014                          .WORD   10000$-.
 6694  034272  005202                          INC     R2              ;NEXT DATA
 6695  034274  005205                          INC     R5              ;NEXT ADDRESS
 6696  034276  022705  000010                  CMP     #10,R5          ;DONE YET?
 6697  034302  001322                          BNE     1$              ;BR IF NO
 6698  034304                                  ENDSEG
 6699  034304                          10000$:
 6700  034304  104405                          TRAP    C$ESEG
 6701  034306                                  EXIT    TST
 6702  034306  104432                          TRAP    C$EXIT
 6703  034310  000012                          .WORD   L10147-.
 6704  034312     000     000     377  5$:     .BYTE   0,0,-1,-1,125,125,252,252
 6705  034315     377     125     125
 6706  034320     252     252
 6707
 6708                                  .EVEN
 6709  034322                          ENDTST
 6710  034322                          L10147:
 6711  034322  104401                          TRAP    C$ETST
 6712
 6713
 6714  034324                                  BADHEAD
 6715                                  ;********************* TEST 63 *********************
 6716                                  ;*ALU TEST
 6717                                  ;*TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT CLEARED
 6718                                  ;*ALU FUNCTION (A-B-1)  CODE=17
 6719                                  ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 6720                                  ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 6721  034324                          BADHEAD
 6722                                  ;********************* TEST 63 *********************
 6723
 6724  034324                          BGNTST
 6725  034324                          T63::
 6726  034324                                  MYINT
 6727  034330                                  MSTCLR                  ;MASTER CLEAR M8200,4,6,7
 6728  034334  005005                          CLR     R5              ;MEM + SP ADDRESS
 6729  034336  012702  034522                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
 6730  034342  004737  003672                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 6731  034346  002642                          MEMDAT                  ;POINTER TO DATA
 6732  034350  004737  004044                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 6733  034354  002652                          SPDAT                   ;POINTER TO DATA
 6734  034356                                  BGNSEG
 6735  034356  104404                          TRAP    C$BSEG
 6736  034360  004737  004112          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
 6737  034364  042737  000017  034402          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 6738  034372  050537  034402                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 6739  034376                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 6740  034402  010000                  2$:     010000                  ;LOAD MAR
 6741  034404  042737  000017  034422          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 6742  034412  050537  034422                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 6743  034416                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 6744  034422  040760                  3$:     040400!<17*20>          ;BR   2'S COMP SUB
 6745  034424                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
6746  034430  061224                          61224                   ;MOVE BR TO PORT4
6747  034432  111237  002612                  MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
6748  034436  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
6749  034442  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
6750  034446  001413                          BEQ     4$              ;BR IS YES
6751  034450                                  ERROR   15              ;ALU ERROR
6752  034466  104455                          TRAP    C$ERDF
6753  034470  000017                          .WORD   15
6754  034472  005223                          .WORD   EM15
6755  034474  010114                          .WORD   ERR15
6756  034476                          4$:     ESCAPE  SEG
6757  034476  104410                          TRAP    C$ESCAPE
6758  034500  000014                          .WORD   10000$-.
6759  034502  005202                          INC     R2              ;NEXT DATA
6760  034504  005205                          INC     R5              ;NEXT ADDRESS
6761  034506  022705  000010                  CMP     #10,R5          ;DONE YET?
6762  034512  001322                          BNE     1$              ;BR IF NO
6763  034514                                  ENDSEG
6764  034514                          10000$:
6765  034514  104405                          TRAP    C$ESEG
6766  034516                                  EXIT    TST
6767  034516  104432                          TRAP    C$EXIT
6768  034520  000012                          .WORD   L10150-.
6769  034522     377     000     376  5$:     .BYTE   -1,0,376,-1,-1,252,124,-1
6770  034525     377     377     252
6771  034530     124     377
6772
6773                                  .EVEN
6774  034532                          ENDTST
6775  034532                          L10150:
6776  034532  104401                          TRAP    C$ETST
6777
6778
6779  034534                                  BADHEAD
6780                                  ;********************* TEST 64 **********************
6781                                  ;*ALU TEST
6782                                  ;*TEST OF ALU FUNCTION DEC A WITH C BIT CLEARED
6783                                  ;*ALU FUNCTION (A-1)     CODE=7
6784                                  ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6785                                  ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
6786
6787  034534                                  BADHEAD
6788                                  ;********************* TEST 64 **********************
6789
6790  034534                          BGNTST
6791  034534                          T64::
6792  034534                                  MYINT
6793  034540                                  MSTCLR                  ;MASTER CLEAR DMC11
6794  034544  005005                          CLR     R5              ;MEM + SP ADDRESS
6795  034546  012702  034732                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
6796  034552  004737  003672                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMMOR
6797  034556  002642                          MEMDAT                  ;POINTER TO DATA
6798  034560  004737  004044                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
6799  034564  002652                          SPDAT                   ;POINTER TO DATA
6800  034566                                  BGNSEG
6801  034566  104404                          TRAP    C$BSEG
```

```
 6802  034570  004737  004112          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
 6803  034574  042737  000017  034612          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 6804  034602  050537  034612                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCITON
 6805  034606                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 6806  034612  010000                  2$:     010000                  ;LOAD MAR
 6807  034614  042737  000017  034632          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 6808  034622  050537  034632                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 6809  034626                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 6810  034632  040560                  3$:     040400!<7*20>           ;BR    DEC A
 6811  034634                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 6812  034640  061224                          61224                   ;MOVE BR TO PORT4
 6813  034642  111237  002612                  MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
 6814  034646  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
 6815  034652  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
 6816  034656  001413                          BEQ     4$              ;BR IF YES
 6817  034660                                  ERROR   15              ;ALU ERROR
 6818  034676  104455                          TRAP    C$ERDF
 6819  034700  000017                          .WORD   15
 6820  034702  005223                          .WORD   EM15
 6821  034704  010114                          .WORD   ERR15
 6822  034706                          4$:     ESCAPE  SEG
 6823  034706  104410                          TRAP    C$ESCAPE
 6824  034710  000014                          .WORD   10000$-.
 6825  034712  005202                          INC     R2              ;NEXT DATA
 6826  034714  005205                          INC      R5             ;NEXT ADDRESS
 6827  034716  022705  000010                  CMP     #10,R5          ;DONE YET?
 6828  034722  001322                          BNE      1$             ;BR IF NO
 6829  034724                                  ENDSEG
 6830  034724                          10000$:
 6831  034724  104405                          TRAP    C$ESEG
 6832  034726                                  EXIT    TST
 6833  034726  104432                          TRAP    C$EXIT
 6834  034730  000012                          .WORD   L10151-.
 6835  034732     377     377     376  5$:     .BYTE   -1,-1,376,376,124,124,251,251
 6836  034735     376     124     124
 6837  034740     251     251
 6838
 6839                                  .EVEN
 6840  034742                          ENDTST
 6841  034742                          L10151:
 6842  034742  104401                          TRAP    C$ETST
 6843
 6844
 6845  034744                                  BADHEAD
 6846                                          ;********************* TEST 65 *********************
 6847                                          ;*ALU TEST
 6848                                          ;*TEST OF ALU FUNCTION SEL B WITH C BIT SET
 6849                                          ;*ALU FUNCTION (B)      CODE=11
 6850                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 6851                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 6852  034744                                  BADHEAD
 6853                                          ;********************* TEST 65 *********************
 6854
 6855  034744                          BGNTST
 6856  034744                          T65::
 6857  034744                                  MYINT
```

```
6858  034750                                  MSTCLR                  ;MASTER CLEAR M8200,4,6,7
6859  034754  005005                          CLR     R5              ;MEM + SP ADDRESS
6860  034756  012702  035142                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
6861  034762  004737  003672                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
6862  034766  002642                          MEMDAT                  ;POINTER TO DATA
6863  034770  004737  004044                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
6864  034774  002652                          SPDAT                   ;POINTET TO DATA
6865  034776                                  BGNSEG
6866  034776  104404                          TRAP    C$BSEG
6867  035000  004737  004130          1$:     JSR     PC,SETC         ;SET C BIT!
6868  035004  042737  000017  035022          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
6869  035012  050537  035022                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
6870  035016                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6871  035022  010000                  2$:     010000                  ;LOAD MAR
6872  035024  042737  000017  035042          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
6873  035032  050537  035042                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
6874  035036                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6875  035042  040620                  3$:     040400!<11*20>          ;BR    SEL B
6876  035044                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6877  035050  061224                          61224                   ;MOVE BR TO PORT4
6878  035052  111237  002612                  MOVB    (R2),$GDDAT             ;PUT 'EXPECTED' IN $GDDAT
6879  035056  116104  000004                  MOVB    4(R1),R4        ;PUT 'FOUND' IN R4
6880  035062  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
6881  035066  001413                          BEQ     4$              ;BR IF YES
6882  035070                                  ERROR   23              ;ALU ERROR
6883  035106  104455                          TRAP    C$ERDF
6884  035110  000027                          .WORD   23
6885  035112  005475                          .WORD   EM23
6886  035114  010624                          .WORD   ERR23
6887  035116                          4$:     ESCAPE  SEG
6888  035116  104410                          TRAP    C$ESCAPE
6889  035120  000014                          .WORD   10000$-.
6890  035122  005202                          INC     R2                      ;NEXT DATA
6891  035124  005205                          INC     R5                      ;NEXT ADDRESS
6892  035126  022705  000010                  CMP     #10,R5                  ;DONE YET?
6893  035132  001322                          BNE     1$                      ;BR IF NO
6894  035134                                  ENDSEG
6895  035134                          10000$:
6896  035134  104405                          TRAP    C$ESEG
6897  035136                                  EXIT    TST
6898  035136  104432                          TRAP    C$EXIT
6899  035140  000012                          .WORD   L10152-.
6900  035142     000     377     000  5$:     .BYTE   0,-1.0,-1,125,252,125,252
6901  035145     377     125     252
6902  035150     125     252
6903
6904
6905  035152                                  .EVEN
6906  035152                                  ENDTST
6907  035152  104401                  L10152: TRAP    C$ETST
6908
6909
6910  035154                                  BADHEAD
6911                                          ;********************* TEST 66 *********************
6912                                          ;*ALU TEST
6913                                          ;*TEST OF ALU FUNCTION SEL A WITH C BIT SET
```

```
                                        ;*ALU FUNCTION (A)      CODE=10
                                        ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                        ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
035154                                  BADHEAD
                                        ;******************** TEST 66 **********************
```

```
6919
6920
6921  035154                                  BGNTST
6922  035154                                  T66::
6923  035154                                          MYINT
6924  035160                                          MSTCLR                  ;MASTER CLEAR M8200,4,6,7
6925  035164  005005                                  CLR     R5              ;MEM + SP ADDRESS
6926  035166  012702  035352                          MOV     #5$,R2          ;POINTER TO CORRECT DATA
6927  035172  004737  003672                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
6928  035176  002642                                  MEMDAT                  ;POINTER TO DATA
6929  035200  004737  004044                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
6930  035204  002652                                  SPDAT                   ;POINTER TO DATA
6931  035206                                          BGNSEG
6932  035206  104404                                  TRAP    C$BSEG
6933  035210  004737  004130              1$:         JSR     PC,SETC         ;SET C BIT!
6934  035214  042737  000017  035232                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
6935  035222  050537  035232                          BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
6936  035226                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6937  035232  010000                      2$:         010000                  ;LOAD MAR
6938  035234  042737  000017  035252                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCITON
6939  035242  050537  035252                          BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
6940  035246                                          ROMCLK                  ;NEXT WORD IS INSTRUCITON, ROMCLK PC=5304
6941  035252  040600                      3$:         040400!<10*20>          ;BR    SEL A
6942  035254                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6943  035260  061224                                  61224                   ;MOVE BR TO PORT4
6944  035262  111237  002612                          MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
6945  035266  116104  000004                          MOVB    4(R1),R4        ;PUT "FOUND" IN R4
6946  035272  123704  002612                          CMPB    $GDDAT,R4               ;DATA CORRECT?
6947  035276  001413                                  BEQ     4$              ;BR IF YES
6948  035300                                          ERROR   23              ;ALU ERROR
6949  035316  104455                                  TRAP    C$ERDF
6950  035320  000027                                  .WORD   23
6951  035322  005475                                  .WORD   EM23
6952  035324  010624                                  .WORD   ERR23
6953  035326                              4$:         ESCAPE  SEG
6954  035326  104410                                  TRAP    C$ESCAPE
6955  035330  000014                                  .WORD   10000$-.
6956  035332  005202                                  INC     R2                      ;NEXT DATA
6957  035334  005205                                  INC     R5                      ;NEXT ADDRESS
6958  035336  022705  000010                          CMP     #10,R5                  ;DONE YET?
6959  035342  001322                                  BNE     1$                      ;BR IF NO
6960  035344                                          ENDSEG
6961  035344                              10000$:
6962  035344  104405                                  TRAP    C$ESEG
6963  035346                                          EXIT    TST
6964  035346  104432                                  TRAP    C$EXIT
6965  035350  000012                                  .WORD   L10153-.
6966  035352     000     000     377     5$:         .BYTE   0,0,-1,-1,125,125,252,252
6967  035355     377     125     125
6968  035360     252     252
6969
6970                                          .EVEN
6971  035362                                  ENDTST
6972  035362                                  L10153:
6973  035362  104401                                  TRAP    C$ETST
6974
```

```
        :
035364                                          BADHEAD
                                                ;******************** TEST 67 ********************
                                                ;*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
035364                                          BADHEAD
                                                ;******************** TEST 67 ********************

035364                                  BGNTST
035364                                  T67::
035364                                          MYINT
035370                                          MSTCLR                          ;MASTER CLEAR M8200,4,6,7
035374  005005                                  CLR     R5                      ;MEM + SP ADDRESS
035376  012702  035562                          MOV     #5$,R2                  ;POINTER TO CORRECT DATA
035402  004737  003672                          JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
035406  002642                                  MEMDAT                          ;POINTER TO DATA
035410  004737  004044                          JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
035414  002652                                  SPDAT                           ;POINTER TO DATA
035416                                          BGNSEG
035416  104404                                  TRAP    C$BSEG
035420  004737  004130                  1$:     JSR     PC,SETC                 ;SET C BIT!
035424  042737  000017  035442                  BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
035432  050537  035442                          BIS     R5,2$                   ;ADD ADDRESS TO INSTRUCTION
035436                                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
035442  010000                          2$:     010000                          ;LOAD MAR
035444  042737  000017  035462                  BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
035452  050537  035462                          BIS     R5,3$                   ;ADD ADDRESS TO INSTRUCTION
035456                                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
035462  040640                          3$:     040400!<12*20>                  ;BR   A OR NOTB
035464                                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
035470  061224                                  61224                           ;MOVE BR TO PORT4
035472  111237  002612                          MOVB    (R2),$GDDAT                     ;PUT "EXPECTED" IN $GDDAT
035476  116104  000004                          MOVB    4(R1),R4                ;PUT "FOUND" IN R4
035502  123704  002612                          CMPB    $GDDAT,R4                       ;DATA CORRECT?
035506  001413                                  BEQ     4$                      ;BR IF YES
035510                                          ERROR    15                     ;ALU ERROR
035526  104455                                  TRAP    C$ERDF
035530  000017                                  .WORD   15
035532  005223                                  .WORD   EM15
035534  010114                                  .WORD   ERR15
035536                                  4$:     ESCAPE  SEG
035536  104410                                  TRAP    C$ESCAPE
035540  000014                                  .WORD   10000$-.
035542  005202                                  INC     R2                      ;NEXT DATA
035544  005205                                  INC     R5                      ;NEXT ADDRESS
035546  022705  000010                          CMP     #10,R5                  ;DONE YET?
035552  001322                                  BNE     1$                      ;BR IF NO
035554                                          ENDSEG
035554                                  10000$:
035554  104405                                  TRAP    C$ESEG
035556                                          EXIT    TST
035556  104432                                  TRAP    C$EXIT
035560  000012                                  .WORD   L10154-.
```

```
7031  035562  377     000     377  5$:     .BYTE   -1,0,-1,-1,-1,125,252,-1
7032  035565  377     377     125
7033  035570  252     377
7034
7035                                .EVEN
7036  035572                        ENDTST
7037  035572                        L10154:
7038  035572  104401                        TRAP    C$ETST
7039
7040
7041  035574                                BADHEAD
7042                                        ;********************* TEST 68 *********************
7043                                        ;*ALU TEST
7044                                        ;*TEST OF ALU FUNCTION A AND B WITH C BIT SET
7045                                        ;ALU FUNCTION (A AND B) CODE=13
7046                                        ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7047                                        ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7048  035574                                BADHEAD
7049                                        ;********************* TEST 68 *********************
7050
7051  035574                        BGNTST
7052  035574                        T68::
7053  035574                                MYINT
7054  035600                                MSTCLR                  ;MASTER CLEAR M8200,4,6,7
7055  035604  005005                        CLR     R5              ;MEM + SP ADDRESS
7056  035606  012702  035772                MOV     #5$,R2          ;POINTER TO CORRECT ADDRESS
7057  035612  004737  003672                JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7058  035616  002642                        MEMDAT                  ;POINTER TO DATA
7059  035620  004737  004044                JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7060  035624  002652                        SPDAT                   ;POINTER TO DATA
7061  035626                                BGNSEG
7062  035626  104404                        TRAP    C$BSEG
7063  035630  004737  004130        1$:     JSR     PC,SETC         ;SET C BIT!
7064  035634  042737  000017  035652        BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7065  035642  050537  035652                BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
7066  035646                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7067  035652  010000                2$:     010000                  ;LOAD MAR
7068  035654  042737  000017  035672        BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7069  035662  050537  035672                BIS     R5,3$           ;ADD ADDRESS TO INSTRUCITON
7070  035666                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7071  035672  040660                3$:     040400!<13*20>          ;BR   A AND B
7072  035674                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7073  035700  061224                        61224                   ;MOVE BR TO PORT4
7074  035702  111237  002612                MOVB    (R2),$GDDAT             ;PUT "EXPECTED"IN $GDDAT
7075  035706  116104  000004                MOVB    4(R1),R4        ;PUT "FOUND" IN R4
7076  035712  123704  002612                CMPB    $GDDAT,R4               ;DATA CORRECT?
7077  035716  001413                        BEQ     4$              ;BR IF YES
7078  035720                                ERROR   23              ;ALU ERROR
7079  035736  104455                        TRAP    C$ERDF
7080  035740  000027                        .WORD   23
7081  035742  005475                        .WORD   EM23
7082  035744  010624                        .WORD   ERR23
7083  035746                        4$:     ESCAPE  SEG
7084  035746  104410                        TRAP    C$ESCAPE
7085  035750  000014                        .WORD   10000$-.
7086  035752  005202                        INC     R2              ;NEXT DATA
```

```
7087  035754  C05205                          INC     R5                      ;NEXT ADDRESS
7088  035756  022705  000010                  CMP     #10,R5                  ;DONE YET?
7089  035762  001322                          BNE     1$                      ;BR IF NO
7090  035764                                  ENDSEG
7091  035764                          10000$:
7092  035764  104405                          TRAP    C$ESEG
7093  035766                                  EXIT    TST
7094  035766  104432                          TRAP    C$EXIT
7095  035770  000012                          .WORD   L10155-.
7096  035772    000     000     000   5$:     .BYTE   0,0,0,-1,125,0,0,252
7097  035775    377     125     000
7098  036000    000     252
7099
7100                                  .EVEN
7101  036002                          ENDTST
7102  036002                          L10155:
7103  036002  104401                          TRAP    C$ETST
7104
7105
7106  036004                                  BADHEAD
7107                                          ;********************* TEST 69 *********************
7108                                          ;*ALU TEST
7109                                          ;*TEST OF ALU FUNCTION A OR B WITH C BIT SET
7110                                          ;*ALU FUNCTION (A OR B) CODE=14
7111                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7112                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7113  036004                                  BADHEAD
7114                                          ;********************* TEST 69 *********************
7115
7116  036004                          BGNTST
7117  036004                          T69::
7118  036004                                  MYINT
7119  036010                                  MSTCLR                          ;MASTER CLEAR M8200,4,6,7
7120  036014  005005                          CLR     R5                      ;MEM + SP ADDRESS
7121  036016  012702  036202                  MOV     #5$,R2                  ;POINTER TO CORRECT DATA
7122  036022  004737  003672                  JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
7123  036026  002642                          MEMDAT                          ;POINTER TO DATA
7124  036030  004737  004044                  JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
7125  036034  002652                          SPDAT                           ;POINTER TO DATA
7126  036036                                  BGNSEG
7127  036036  104404                          TRAP    C$BSEG
7128  036040  004737  004130          1$:     JSR     PC,SETC                 ;SET C BIT!
7129  036044  042737  000017  036062          BIC #17,2$                      ;CLEAR ADDRESS FIELD OF INSTRUCTION
7130  036052  050537  036062                  BIS     R5,2$                   ;ADD ADDRESS TO INSTRUCTION
7131  036056                                  ROMCLK                          ;NEXT WORD IS INSTRUCITON, ROMCLK PC=5304
7132  036062  010000                  2$:     010000                          ;LOAD MAR
7133  036064  042737  000017  036102          BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
7134  036072  050537  036102                  BIS     R5,3$                   ;ADD ADDRESS TO INSTRUCTION
7135  036076                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7136  036102  040700                  3$:     040400!<14*20>                  ;BR   A OR B
7137  036104                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7138  036110  061224                          61224                           ;MOVE BR TO PORT4
7139  036112  111237  002612                  MOVB    (R2),$GDDAT                     ;PUT 'EXPECTED' IN R4
7140  036116  116104  000004                  MOVB    4(R1),R4                ;PUT 'FOUND' IN R4
7141  036122  123704  002612                  CMPB    $GDDAT,R4                       ;DATA CORRECT?
7142  036126  001413                          BEQ     4$                      ;BR IF YES
```

```
036130                                      ERROR   23              ;ALU ERROR
036146  104455                              TRAP    C$ERDF
036150  000027                              .WORD   23
036152  005475                              .WORD   EM23
036154  010624                              .WORD   ERR23
036156                              4$:     ESCAPE  SEG
036156  104410                              TRAP    C$ESCAPE
036160  000014                              .WORD   10000$-.
036162  005202                              INC     R2              ;NEXT DATA
036164  005205                              INC     R5              ;NEXT ADDRESS
036166  022705  000010                      CMP     #10,R5          ;DONE YET?
036172  001322                              BNE     1$              ;BR IF NO
036174                                      ENDSEG
036174                              10000$:
036174  104405                              TRAP    C$ESEG
036176                                      EXIT    TST
036176  104432                              TRAP    C$EXIT
036200  000012                              .WORD   L10156-.
036202     000     377     377      5$:     .BYTE   0,-1,-1,-1,125,-1,-1,252
036205     377     125     377
036210     377     252


                                    .EVEN
036212                              ENDTST
036212                              L10156:
036212  104401                              TRAP    C$ETST


036214                                      BADHEAD
                                            ;******************** TEST 70 ********************
                                            ;*ALU TEST
                                            ;*TEST OF ALU FUNCTION A XOR B WITH C BIT SET
                                            ;*ALU FUNCTION (A XOR B)        CODE=15
                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                                    ;*PERFORM THE FUNCTION, VERIFY THE RESULTS

036214                                      BADHEAD
                                            ;******************** TEST 70 ********************
036214                              BGNTST
036214                              T70::
036214                                      MYINT
036220                                      MSTCLR                  ;MASTER CLEAR M8200,4,6,7
036224  005005                              CLR     R5              ;MEM + SP ADDRESS
036226  012702  036412                      MOV     #5$,R2          ;POINTER TO CORRECT DATA
036232  004737  003672                      JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
036236  002642                              MEMDAT                  ;POINTER TO DATA
036240  004737  004044                      JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
036244  002652                              SPDAT                   ;POINTER TO DATA
036246                                      BGNSEG
036246  104404                              TRAP    C$BSEG
036250  004737  004130              1$:     JSR     PC,SETC         ;SET C BIT!
036254  042737  000017  036272              BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
036262  050537  036272                      BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
036266                                      ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
036272  010000                      2$:     010000                  ;LOAD MAR
036274  042737  000017  036312              BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
```

```
 7199  036302  050537  036312                BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 7200  036306                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7201  036312  040720                3$:     040400!<15*20>          ;BR   A XOR B
 7202  036314                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7203  036320  061224                        61224                   ;MOVE BR TO PORT4
 7204  036322  111237  002612                MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
 7205  036326  116104  000004                MOVB    4(R1),R4        ;PUT "FOUND" IN R4
 7206  036332  123704  002612                CMPB    $GDDAT,R4               ;DATA CORRECT?
 7207  036336  001413                        BEQ     4$              ;BR IF YES
 7208  036340                                ERROR   23              ;ALU ERROR
 7209  036356  104455                        TRAP    C$ERDF
 7210  036360  000027                        .WORD   23
 7211  036362  005475                        .WORD   EM23
 7212  036364  010624                        .WORD   ERR23
 7213  036366                        4$:     ESCAPE  SEG
 7214  036366  104410                        TRAP    C$ESCAPE
 7215  036370  000014                        .WORD   10000$-.
 7216  036372  005202                        INC     R2              ;NEXT DATA
 7217  036374  005205                        INC     R5              ;NEXT ADDRESS
 7218  036376  022705  000010                CMP     #10,R5          ;DONE YET?
 7219  036402  001322                        BNE     1$              ;BR IF NO
 7220  036404                                ENDSEG
 7221  036404                        10000$:
 7222  036404  104405                        TRAP    C$ESEG
 7223  036406                                EXIT    TST
 7224  036406  104432                        TRAP    C$EXIT
 7225  036410  000012                        .WORD   L10157-.
 7226  036412     000     377     377 5$:    .BYTE   0,-1,-1,0,0,-1,-1,0
 7227  036415     000     000     377
 7228  036420     377     000
 7229
 7230                                .EVEN
 7231  036422                        ENDTST
 7232  036422                        L10157:
 7233  036422  104401                        TRAP    C$ETST
 7234
 7235
 7236  036424                                BADHEAD
 7237                                        ;********** TEST 71 **********
 7238                                        ;*ALU TEST
 7239                                        ;*TEST OF ALU FUNCTION ADD WITH C BIT SET
 7240                                        ;*ALU FUNCTION (A PLUS B)       CODE=00
 7241                                        ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 7242                                        ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 7243  036424                                BADHEAD
 7244                                        ;********** TEST 71 **********
 7245
 7246  036424                        BGNTST
 7247  036424                        T71::
 7248  036424
 7249  036430                                MYINT
 7250  036434  005005                        MSTCLR                  ;MASTER CLEAR M8200,4,6,7
 7251  036436  012702  036622                CLR     R5              ;MEM + SP ADDRESS
 7252  036442  004737  003672                MOV     #5$,R2          ;POINTER TO CORRECT DATA
 7253  036446  002642                        JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 7254  036450  004737  004044                MEMDAT                  ;POINTER TO DATA
                                             JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
```

```
7255  036454  002652                          SPDAT                   ;OOINTER TO DATA
7256  036456                                  BGNSEG
7257  036456  104404                          TRAP    C$BSEG
7258  036460  004737  004130          1$:     JSR     PC,SETC         ;SET C BIT!
7259  036464  042737  000017  036502        BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7260  036472  050537  036502                BIS     R5,2$           ;ADD ADDRESS TO INSTRUCITON
7261  036476                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7262  036502  010000                  2$:     010000                  ;LOAD MAR
7263  036504  042737  000017  036522        BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7264  036512  050537  036522                BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
7265  036516                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7266  036522  040400                  3$:     040400!<00*20>          ;BR   ADD
7267  036524                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7268  036530  061224                          61224                   ;MOVE BR TO PORT4
7269  036532  111237  002612                MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
7270  036536  116104  000004                MOVB    4(R1),R4        ;PUT "FOUND" IN R4
7271  036542  123704  002612                CMPB    $GDDAT,R4               ;DATA CORRECT?
7272  036546  001413                        BEQ     4$              ;BR IF YES
7273  036550                                  ERROR   23              ;ALU ERROR
7274  036566  104455                          TRAP    C$ERDF
7275  036570  000027                          .WORD   23
7276  036572  005475                          .WORD   EM23
7277  036574  010624                          .WORD   ERR23
7278  036576                          4$:     ESCAPE  SEG
7279  036576  104410                          TRAP    C$ESCAPE
7280  036600  000014                          .WORD   10000$-.
7281  036602  005202                          INC     R2              ;NEXT DATA
7282  036604  005205                          INC     R5              ;NEXT ADDRESS
7283  036606  022705  000010                CMP     #10,R5          ;DONE YET?
7284  036612  001322                        BNE     1$              ;BR IF NO
7285  036614                                  ENDSEG
7286  036614                          10000$:
7287  036614  104405                          TRAP    C$ESEG
7288  036616                                  EXIT    TST
7289  036616  104432                          TRAP    C$EXIT
7290  036620  000012                          .WORD   L10160-.
7291  036622     000  377  377  5$:     .BYTE   0,-1,-1,376,252,-1,-1,124
7292  036625     376  252  377
7293  036630     377  124
7294
7295                                  .EVEN
7296  036632                          ENDTST
7297  036632                          L10160:
7298  036632  104401                          TRAP    C$ETST
7299
7300
7301  036634                                  BADHEAD
7302                                          ;********** TEST 72 **********
7303                                          ;*ALU TEST
7304                                          ;*TEST OF ALU FUNCTION 2A W/C WITH C BIT SET
7305                                          ;*ALU FUNCTION (A PLUS A PLUS C)        CODE=6
7306                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7307                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7308  036634                                  BADHEAD
7309                                          ;********** TEST 72 **********
7310
```

```
 7311  036634                                  BGNTST
 7312  036634                                  T72::
 7313  036634                                        MYINT
 7314  036640                                        MSTCLR                          ;MASTER CLEAR M8200,4,6,7
 7315  036644  005005                                CLR     R5                      ;MEM + SP ADDRESS
 7316  036646  012702  037032                        MOV     #5$,R2                  ;POINTER TO CORRECT DATA
 7317  036652  004737  003672                        JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
 7318  036656  002642                                MEMDAT                          ;POINTER TO DATA
 7319  036660  004737  004044                        JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
 7320  036664  002652                                SPDAT                           ;POINTER TO DATA
 7321  036666                                        BGNSEG
 7322  036666  104404                                TRAP    C$BSEG
 7323  036670  004737  004130              1$:       JSR     PC,SETC                 ;SET C BIT!
 7324  036674  042737  000017  036712                BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7325  036702  050537  036712                        BIS     R5,2$                   ;ADD ADDRESS TO INSTRUCTION
 7326  036706                                        ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7327  036712  010000                        2$:     010000                          ;LOAD MAR
 7328  036714  042737  000017  036732                BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
 7329  036722  050537  036732                        BIS     R5,3$                   ;ADD ADDRESS TO INSTRUCTION
 7330  036726                                        ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=55304
 7331  036732  040540                        3$:     040400!<6*20>                   ;BR    2A W/C
 7332  036734                                        ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7333  036740  061224                                61224                           ;MOVE BR TO PORT4
 7334  036742  111237  002612                        MOVB    (R2),$GDDAT                     ;PUT 'WXPECTED' IN $GDDAT
 7335  036746  116104  000004                        MOVB    4(R1),R4                ;PUT 'FOUND' IN R4
 7336  036752  123704  002612                        CMPB    $GDDAT,R4                       ;DATA CORRECT?
 7337  036756  001413                                BEQ     4$                      ;BR IF YES
 7338  036760                                        ERROR   23                      ;ALU ERROR
 7339  036776  104455                                TRAP    C$ERDF
 7340  037000  000027                                .WORD   23
 7341  037002  005475                                .WORD   EM23
 7342  037004  010624                                .WORD   ERR23
 7343  037006                                4$:     ESCAPE  SEG
 7344  037006  104410                                TRAP    C$ESCAPE
 7345  037010  000014                                .WORD   10000$-.
 7346  037012  005202                                INC     R2                      ;NEXT DATA
 7347  037014  005205                                INC     R5                      ;NEXT ADDRESS
 7348  037016  022705  000010                        CMP     #10,R5                  ;DONE YET?
 7349  037022  001322                                BNE     1$                      ;BR IF NO
 7350  037024                                        ENDSEG
 7351  037024                                10000$:
 7352  037024  104405                                TRAP    C$ESEG
 7353  037026                                        EXIT    TST
 7354  037026  104432                                TRAP    C$EXIT
 7355  037030  000012                                .WORD   L10161-.
 7356  037032     001     001     377        5$:     .BYTE   1,1,-1,-1,253,253,125,125
 7357  037035     377     253     253
 7358  037040     125     125
 7359
 7360
 7361  037042                                .EVEN
 7362  037042                                ENDTST
 7363  037042  104401                        L10161:
 7364                                                TRAP    C$ETST
 7365
 7366  037044                                        BADHEAD
```

```
                                              ;******************** TEST 73 ********************
                                              ;*ALU TEST
                                              ;*TEST OF ALU FUNCTION SUB WITH C BIT SET
                                              ;*ALU FUNCTION (A-B)    CODE=16
                                              ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                              ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
037044                                        BADHEAD
                                              ;******************** TEST 73 ********************

037044                               BGNTST
037044                               T73::
037044                                        MYINT
037050                                        MSTCLR                        ;MASTER CLEAR M8200,4,6,7
037054  005005                                CLR     R5                    ;MEM + SP ADDRESS
037056  012702  037242                        MOV #5$,R2                    ;POINTER TO CORRECT DATA
037062  004737  003672                        JSR     PC,MEMLD              ;LOAD 8 WORDS OF MAIN MEMORY
037066  002642                                MEMDAT                        ;POINTER TO DATA
037070  004737  004044                        JSR     PC,SPLD               ;LOAD 8 WORDS OF SP
037074  002652                                SPDAT                         ;POINTER TO DATA
037076                                        BGNSEG
037076  104404                                TRAP    C$BSEG
037100  004737  004130               1$:      JSR     PC,SETC               ;SET C BIT!
037104  042737  000017  037122                BIC     #17,2$                ;CLEAR ADDRESS FIELD OF INSTRUCTION
037112  050537  037122                        BIS     R5,2$                 ;ADD ADDRESS TO INSTRUCTION
037116                                        ROMCLK                        ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
037122  010000                       2$:      010000                        ;LOAD MAR
037124  042737  000017  037142                BIC     #17,3$                ;CLEAR ADDRESS OF INSTRUCTION
037132  050537  037142                        BIS     R5,3$                 ;ADD ADDRESS  TO INSTRUCTION
037136                                        ROMCLK                        ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
037142  040740                       3$:      040400!<16*20>                ;BR   SUB
037144                                        ROMCLK                        ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
037150  061224                                61224                         ;MOVE BR TO PORT4
037152  111237  002612                        MOVB    (R2),$GDDAT                   ;PUT "EXPECTED" IN $GDDAT
037156  116104  000004                        MOVB    4(R1),R4              ;PUT "FOUND" IN R4
037162  123704  002612                        CMPB    $GDDAT,R4                     ;DATA CORRECT?
037166  001413                                BEQ     4$                    ;BR IF YES
037170                                        ERROR   23                    ;ALU ERROR
037206  104455                                TRAP    C$ERDF
037210  000027                                .WORD   23
037212  005475                                .WORD   EM23
037214  010624                                .WORD   ERR23
037216                               4$:      ESCAPE  SEG
037216  104410                                TRAP    C$ESCAPE
037220  000014                                .WORD   10000$-.
037222  005202                                INC     R2                    ;NEXT DATA
037224  005205                                INC     R5                    ;NEXT ADDRESS
037226  022705  000010                        CMP     #10,R5                ;DONE YET?
037232  001322                                BNE     1$                    ;BR IF NO
037234                                        ENDSEG
037234                               10000$:
037234  104405                                TRAP    C$ESEG
037236                                        EXIT    TST
037236  104432                                TRAP    C$EXIT
037240  000012                                .WORD   L10162-.
037242     000     001     377       5$:      .BYTE   0,1,-1,0,0,253,125,0
037245     000     000     253
```

```
 7423  037250  125  000
 7424
 7425                                .EVEN
 7426  037252                        ENDTST
 7427  037252                        L10162:
 7428  037252  104401                        TRAP    C$ETST
 7429
 7430
 7431  037254                                BADHEAD
 7432                                ;********************* TEST 74 *********************
 7433                                ;*ALU TEST
 7434                                ;*TEST OF ALU FUNCTION ADD W/C WITH C BIT SET
 7435                                ;*ALU FUNCTION (A PLUS B PLUS C)        CODE=01
 7436                                ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 7437                                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 7438  037254                        BADHEAD
 7439                                ;********************* TEST 74 *********************
 7440
 7441  037254                        BGNTST
 7442  037254                        T74::
 7443  037254                                MYINT
 7444  037260                                MSTCLR                          ;MASTER CLEAR M8200,4,6,7
 7445  037264  005005                        CLR     R5                      ;MEM +SP ADDRESS
 7446  037266  012702  037452                MOV     #5$,R2                  ;POINTER TO CORRECT DATA
 7447  037272  004737  003672                JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
 7448  037276  002642                        MEMDAT                          ;POINTER TO DATA
 7449  037300  004737  004044                JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
 7450  037304  002652                        SPDAT                           ;POINTER TO DATA
 7451  037306                                BGNSEG
 7452  037306  104404                        TRAP    C$BSEG
 7453  037310  004737  004130        1$:     JSR     PC,SETC                 ;SET C BIT!
 7454  037314  042737  000017  037332        BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7455  037322  050537  037332                BIS     R5,2$                   ;ADD ADDRESS TO INSTRUCTION
 7456  037326                                ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7457  037332  010000                2$:     010000                          ;LOAD MAR
 7458  037334  042737  000017  037352        BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
 7459  037342  050537  037352                BIS     R5,3$                   ;ADD ADDRESS TO INSTRUCTION
 7460  037346                                ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7461  037352  040420                3$:     040400!<01*20>                  ;BR - ADD W/C
 7462  037354                                ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7463  037360  061224                        61224                           ;MOVE BR TO PORT4
 7464  037362  111237  002612                MOVB    (R2),$GDDAT                     ;PUT "EXPECTED" IN $GDDAT
 7465  037366  116104  000004                MOVB    4(R1),R4                ;PUT "FOUND" IN R4
 7466  037372  123704  002612                CMPB    $GDDAT,R4                       ;DATA CORRECT?
 7467  037376  001413                        BEQ     4$                      ;BR IF YES
 7468  037400                                ERROR   23                      ;ALU ERROR
 7469  037416  104455                        TRAP    C$ERDF
 7470  037420  000027                        .WORD   23
 7471  037422  005475                        .WORD   EM23
 7472  037424  010624                        .WORD   ERR23
 7473  037426                        4$:     ESCAPE  SEG
 7474  037426  104410                        TRAP    C$ESCAPE
 7475  037430  000014                        .WORD   10000$-.
 7476  037432  005202                        INC     R2                      ;NEXT DATA
 7477  037434  005205                        INC     R5                      ;NEXT ADDRESS
 7478  037436  022705  000010                CMP     #10,R5                  ;DONE YET?
```

```
7479  037442  001322                          BNE     1$              ;BR IF NO
7480  037444                                  ENDSEG
7481  037444                          10000$:
7482  037444  104405                          TRAP    C$ESEG
7483  037446                                  EXIT    TST
7484  037446  104432                          TRAP    C$EXIT
7485  037450  000012                          .WORD   L10163-.
7486  037452     001     000     000  5$:     .BYTE   1,0,0,-1,253,0,0,125
7487  037455     377     253     000
7488  037460     000     125
7489
7490                                  .EVEN
7491  037462                          ENDTST
7492  037462                          L10163:
7493  037462  104401                          TRAP    C$ETST
7494
7495
7496  037464                                  BADHEAD
7497                                          ;******************** TEST 75 **********************
7498                                          ;*ALU TEST
7499                                          ;*TEST OF ALU FUNCTION SUB W/C WITH C BIT SET
7500                                          ;*ALU FUNCTION (A-B-C)  CODE=2
7501                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7502                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7503  037464                                  BADHEAD
7504                                          ;******************** TEST 75 **********************
7505
7506
7507  037464                          BGNTST
7508  037464                          T75::
7509  037464                                  MYINT
7510  037470                                  MSTCLR
7511  037474  005005                          CLR     R5              ;MASTER CLEAR M8200,4,6,7
7512  037476  012702  037662                  MOV     #5$,R2          ;MEM + SP ADDRESS
7513  037502  004737  003672                  JSR     PC,MEMLD        ;POINTER TO CORRECT DATA
7514  037506  002642                          MEMDAT                  ;LOAD 8 WORDS OF MAIN MEMORY
7515  037510  004737  004044                  JSR     PC,SPLD         ;POINTER TO DATA
7516  037514  002652                          SPDAT                   ;LOAD 8 WORDS OF SP
7517  037516                                  BGNSEG                  ;POINTER TO DATA
7518  037516  104404                          TRAP    C$BSEG
7519  037520  004737  004130          1$:     JSR     PC,SETC         ;SET C BIT!
7520  037524  042737  000017  037542          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7521  037532  050537  037542                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
7522  037536                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7523  037542  010000                  2$:     010000                  ;LOAD MAR
7524  037544  042737  000017  037562          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7525  037552  050537  037562                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
7526  037556                                  ROMCLK                  ;NEXT WORD IS INSTRUCITON, ROMCLK PC=55304
7527  037562  040440                  3$:     040400!<2*20>           ;BR   SUB W/C
7528  037564                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7529  037570  061224                          61224                   ;MOVE BR TO PORT4
7530  037572  111237  002612                  MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
7531  037576  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
7532  037602  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
7533  037606  001413                          BEQ     4$              ;BR IF YES
7534  037610                                  ERROR   23              ;ALU ERROR
```

```
7535  037626  104455                          TRAP    C$ERDF
7536  037630  000027                          .WORD   23
7537  037632  005475                          .WORD   EM23
7538  037634  010624                          .WORD   ERR23
7539  037636                          4$:     ESCAPE  SEG
7540  037636  104410                          TRAP    C$ESCAPE
7541  037640  000014                          .WORD   10000$-.
7542  037642  005202                          INC     R2              ;NEXT DATA
7543  037644  005205                          INC     R5              ;NEXT ADDRESS
7544  037646  022705  000010                  CMP     #10,R5                  ;DONE YET?
7545  037652  001322                          BNE     1$              ;BR IF NO
7546  037654                                  ENDSEG
7547  037654                          10000$:
7548  037654  104405                          TRAP    C$ESEG
7549  037656                                  EXIT    TST
7550  037656  104432                          TRAP    C$EXIT
7551  037660  000012                          .WORD   L10164-.
7552  037662     000     001     377  5$:     .BYTE   0,1,-1,0,0,253,125,0
7553  037665     000     000     253
7554  037670     125     000
7555
7556                                  .EVEN
7557  037672                          ENDTST
7558  037672                          L10164:
7559  037672  104401                          TRAP    C$ETST
7560
7561
7562  037674                                  BADHEAD
7563                                  ;********************* TEST 76 *********************
7564                                  ;*ALU TEST
7565                                  ;*TEST OF ALU FUNCTION INC A WITH C  BIT SET
7566                                  ;*ALU FUNCTION (A PLUS 1)       CODE=3
7567                                  ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7568                                  ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7569  037674                          BADHEAD
7570                                  ;********************* TEST 76 *********************
7571
7572  037674                  BGNTST
7573  037674                  T76::
7574  037674                          MYINT
7575  037700                          MSTCLR                          ;MASTER CLEAR M8200,4,6,7
7576  037704  005005                  CLR     R5                      ;MEM + SP ADDRESS
7577  037706  012702  040072          MOV     #5$,R2                  ;POINTER TO CORRECT DATA
7578  037712  004737  003672          JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
7579  037716  002642                  MEMDAT                          ;POINTER TO DATA
7580  037720  004737  004044          JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
7581  037724  002652                  SPDAT                           ;POINTER TO DATA
7582  037726                          BGNSEG
7583  037726  104404                  TRAP    C$BSEG
7584  037730  004737  004130  1$:     JSR     PC,SETC                 ;SET C BIT!
7585  037734  042737  000017  037752  BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
7586  037742  050537  037752          BIS     R5,2$                   ;ADD ADDRESS TO INSTRUCTION
7587  037746                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7588  037752  010000          2$:     010000                          ;LOAD MAR
7589  037754  042737  000017  037772  BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
7590  037762  050537  037772          BIS     R5,3$                   ;ADD ADDRESS TO INSTRUCTION
```

```
7591  037766
7592  037772  040460                3$:     ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7593  037774                                040400!<3*20>                   ;BR   INC A
7594  040000  061224                       ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7595  040002  111237  002612               61224                           ;MOVE BR TO PORT4
7596  040006  116104  000004               MOVB    (R2),$GDDAT                   ;PUT "EXPECTED" IN $GDDAT
7597  040012  123704  002612               MOVB    4(R1),R4                ;PUT "FOUND IN R4
7598  040016  001413                       CMPB    $GDDAT,R4                     ;DATA CORRECT?
7599  040020                                BEQ     4$                      ;BR IF YES
7600  040036  104455                       ERROR   23                      ;ALU ERROR
7601  040040  000027                       TRAP    C$ERDF
7602  040042  005475                       .WORD   23
7603  040044  010624                       .WORD   EM23
7604  040046                               .WORD   ERR23
7605  040046  104410                4$:    ESCAPE  SEG
7606  040050  000014                       TRAP    C$ESCAPE
7607  040052  005202                       .WORD   10000$-.
7608  040054  005205                       INC     R2                      ;NEXT DATA
7609  040056  022705  000010               INC     R5                      ;NEXT ADDRESS
7610  040062  001322                       CMP     #10,R5                  ;DONE YET?
7611  040064                               BNE     1$                      ;BR IF NO
7612  040064                               ENDSEG
7613  040064  104405                10000$:
7614  040066                               TRAP    C$ESEG
7615  040066  104432                       EXIT    TST
7616  040070  000012                       TRAP    C$EXIT
7617  040072    001     001     000 5$:    .WORD   L10165-.
7618  040075    000     126     126        .BYTE   1,1,0,0,126,126,253,253
7619  040100    253     253
7620
7621                                 .EVEN
7622  040102                         ENDTST
7623  040102                         L10165:
7624  040102  104401                       TRAP    C$ETST
7625
7626
7627  040104                               BADHEAD
7628                                       ;********************* TEST 77 *********************
7629                                       ;*ALU TEST
7630                                       ;*TEST OF ALU FUNCTION 2A WITH C BIT SET
7631                                       ;*ALU FUNCTION (A PLUS A)       CODE=5
7632                                       ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7633                                       ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7634  040104                               BADHEAD
7635                                       ;********************* TEST 77 *********************
7636
7637  040104                         BGNTST
7638  040104                         T77::
7639  040104
7640  040110                               MYINT
7641  040114  005005                       MSTCLR                          ;MASTER CLEAR M8200,4,6,7
7642  040116  012702  040302               CLR     R5                      ;MEM + SP ADDRESS
7643  040122  004737  003672               MOV     #5$,R2                  ;POINTER TO CORRECT DATA
7644  040126  002642                       JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
7645  040130  004737  004044               MEMDAT                          ;POINTER TO DATA
7646  040134  002652                       JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
                                           SPDAT                           ;POINTER TO DATA
```

```
 7647  040136                                    BGNSEG
 7648  040136  104404                            TRAP    C$BSEG
 7649  040140  004737  004130              1$:   JSR     PC,SETC         ;SET C BIT!
 7650  040144  042737  000017  040162          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 7651  040152  050537  040162                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
 7652  040156                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7653  040162  010000                      2$:   010000                  ;LOAD MAR
 7654  040164  042737  000017  040202          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 7655  040172  050537  040202                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
 7656  040176                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7657  040202  040520                      3$:   040400!<5*20>           ;BR    2A
 7658  040204                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 7659  040210  061224                            61224                   ;MOVE BR TO PORT4
 7660  040212  111237  002612                    MOVB    (R2),$GDDAT             ;PUT ''EXPECTED'' IN $GDDAT
 7661  040216  116104  000004                    MOVB    4(R1),R4        ;PUT ''FOUND IN R4
 7662  040222  123704  002612                    CMPB    $GDDAT,R4               ;DATA CORRECT?
 7663  040226  001413                            BEQ     4$              ;BR IF YES
 7664  040230                                    ERROR   23              ;ALU ERROR
 7665  040246  104455                            TRAP    C$ERDF
 7666  040250  000027                            .WORD   23
 7667  040252  005475                            .WORD   EM23
 7668  040254  010624                            .WORD   ERR23
 7669  040256                              4$:   ESCAPE  SEG
 7670  040256  104410                            TRAP    C$ESCAPE
 7671  040260  000014                            .WORD   10000$-.
 7672  040262  005202                            INC     R2              ;NEXT DATA
 7673  040264  005205                            INC     R5              ;NEXT ADDRESS
 7674  040266  022705  000010                    CMP     #10,R5          ;DONE YET?
 7675  040272  001322                            BNE     1$              ;BR IF NO
 7676  040274                                    ENDSEG
 7677  040274                            10000$:
 7678  040274  104405                            TRAP    C$ESEG
 7679  040276                                    EXIT    TST
 7680  040276  104432                            TRAP    C$EXIT
 7681  040300  000012                            .WORD   L10166-.
 7682  040302     000     000     376    5$:     .BYTE   0,0,376,376,252,252,124,124
 7683  040305     376     252     252
 7684  040310     124     124
 7685
 7686                                      .EVEN
 7687  040312                              ENDTST
 7688  040312                              L10166:
 7689  040312  104401                            TRAP    C$ETST
 7690
 7691
 7692  040314                                    BADHEAD
 7693                                            ;********************* TEST 78 *********************
 7694                                            ;*ALU TEST
 7695                                            ;*TEST OF ALU FUNCTION A PLUS C WITH C BIT SET
 7696                                            ;*ALU FUNCTION (A PLUS C)       CODE=4
 7697                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 7698                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 7699  040314                                    BADHEAD
 7700                                            ;********************* TEST 78 *********************
 7701
 7702  040314                              BGNTST
```

```
7703  040314                          T78::
7704  040314                                MYINT
7705  040320                                MSTCLR                  ;MASTER CLEAR M8200,4,6,7
7706  040324  005005                        CLR     R5              ;MEM + SP ADDRESS
7707  040326  012702  040512                MOV     #5$,R2          ;POINTER TO CORRECT DATA
7708  040332  004737  003672                JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7709  040336  002642                        MEMDAT                  ;POINTER TO DATA
7710  040340  004737  004044                JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
7711  040344  002652                        SPDAT                   ;POINTER TO DATA
7712  040346                                BGNSEG
7713  040346  104404                        TRAP    C$BSEG
7714  040350  004737  004130          1$:   JSR     PC,SETC         ;SET C BIT!
7715  040354  042737  000017  040372        BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7716  040362  050537  040372                BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
7717  040366                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7718  040372  010000                  2$:   010000                  ;LOAD MAR
7719  040374  042737  000017  040412        BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7720  040402  050537  040412                BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
7721  040406                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7722  040412  040500                  3$:   040400!<4*20>           ;BR   A PLUS C
7723  040414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7724  040420  061224                        61224                   ;MOVE BR TO PORT4
7725  040422  111237  002612                MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
7726  040426  116104  000004                MOVB    4(R1),R4        ;PUT "FOUND IN R4
7727  040432  123704  002612                CMPB    $GDDAT,R4               ;DATA CORRECT?
7728  040436  001413                        BEQ     4$              ;BR IF YES
7729  040440                                ERROR   23              ;ALU ERROR
7730  040456  104455                        TRAP    C$ERDF
7731  040460  000027                        .WORD   23
7732  040462  005475                        .WORD   EM23
7733  040464  010624                        .WORD   ERR23
7734  040466                          4$:   ESCAPE  SEG
7735  040466  104410                        TRAP    C$ESCAPE
7736  040470  000014                        .WORD   10000$-.
7737  040472  005202                        INC     R2                      ;NEXT DATA
7738  040474  005205                        INC     R5                      ;NEXT ADDRESS
7739  040476  022705  000010                CMP     #10,R5                  ;DONE YET?
7740  040502  001322                        BNE     1$                      ;BR IF NO
7741  040504                                ENDSEG
7742  040504                          10000$:
7743  040504  104405                        TRAP    C$ESEG
7744  040506                                EXIT    TST
7745  040506  104432                        TRAP    C$EXIT
7746  040510  000012                        .WORD   L10167-.
7747  040512     001     001     000  5$:   .BYTE   1,1,0,0,126,126,253,253
7748  040515     000     126     126
7749  040520     253     253
7750
7751                                .EVEN
7752  040522                        ENDTST
7753  040522                        L10167:
7754  040522  104401                        TRAP    C$ETST
7755
7756  040524                                BADHEAD
7757                                        ;********************* TEST 79 *********************
7758                                        ;*ALU TEST
```

```
 7759                                   ;*TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT SET
 7760                                   ;*ALU FUNCTION (A-B-1)  CODE=17
 7761                                   ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 7762                                   ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 7763  040524                           BADHEAD
 7764                                   ;********************* TEST 79 **********************
 7765
 7766  040524                   BGNTST
 7767  040524                   T79::
```

```
7768  040524                                  MYINT
7769  040530                                  MSTCLR
7770  040534  005005                          CLR     R5              ;MASTER CLEAR M8200,4,6,7
7771  040536  012702  040722                  MOV     #5$,R2          ;MEM + SP ADDRESS
7772  040542  004737  003672                  JSR     PC,MEMLD        ;POINTER TO CORRECT DATA
7773  040546  002642                          MEMDAT                  ;LOAD 8 WORDS OF MAIN MEMORY
7774  040550  004737  004044                  JSR     PC,SPLD         ;POINTER TO DATA
7775  040554  002652                          SPDAT                   ;LOAD 8 WORDS OF SP
7776  040556                                  BGNSEG                  ;POINTER TO DATA
7777  040556  104404                          TRAP    C$BSEG
7778  040560  004737  004130          1$:     JSR     PC,SETC         ;SET C BIT!
7779  040564  042737  000017  040602          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
7780  040572  050537  040602                  BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
7781  040576                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7782  040602  010000                  2$:     010000                  ;LOAD MAR
7783  040604  042737  000017  040622          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
7784  040612  050537  040622                  BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
7785  040616                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7786  040622  040760                  3$:     040400!<17*20>          ;BR   2'S COMP SUB
7787  040624                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7788  040630  061224                          61224                   ;MOVE BR TO PORT4
7789  040632  111237  002612                  MOVB    (R2),$GDDAT             ;PUT "EXPECTED" IN $GDDAT
7790  040636  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND IN R4
7791  040642  123704  002612                  CMPB    $GDDAT,R4               ;DATA CORRECT?
7792  040646  001413                          BEQ     4$              ;BR IF YES
7793  040650                                  ERROR   23              ;ALU ERROR
7794  040666  104455                          TRAP    C$ERDF
7795  040670  000027                          .WORD   23
7796  040672  005475                          .WORD   EM23
7797  040674  010624                          .WORD   ERR23
7798  040676                          4$:     ESCAPE  SEG
7799  040676  104410                          TRAP    C$ESCAPE
7800  040700  000014                          .WORD   10000$-.
7801  040702  005202                          INC     R2              ;NEXT DATA
7802  040704  005205                          INC     R5              ;NEXT ADDRESS
7803  040706  022705  000010                  CMP     #10,R5          ;DONE YET?
7804  040712  001322                          BNE     1$              ;BR IF NO
7805  040714                                  ENDSEG
7806  040714                          10000$:
7807  040714  104405                          TRAP    C$ESEG
7808  040716                                  EXIT    TST
7809  040716  104432                          TRAP    C$EXIT
7810  040720  000012                          .WORD   L10170-.
7811  040722     377     000     376  5$:     .BYTE   -1,0,376,-1,-1,252,124,-1
7812  040725     377     377     252
7813  040730     124     377
7814
7815                                  .EVEN
7816  040732                          ENDTST
7817  040732                          L10170:
7818  040732  104401                          TRAP    C$ETST
7819
7820
7821  040734                                  BADHEAD
7822                                          ;********************* TEST 80 *********************
7823                                          ;*ALU TEST
```

```
                                                ;*TEST OF ALU FUNCTION DEC A WITH C BIT SET
                                                ;*ALU FUNCTION (A-1)    CODE=7
                                                ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
040734                                          BADHEAD
                                                ;******************** TEST 80 **********************

040734                                  BGNTST
040734                                  T80::
040734                                          M.INT
040740                                          MSTCLR                  ;MASTER CLEAR M8200,4,6,7
040744  005005                                  CLR     R5              ;MEM + SP ADDRESS
040746  012702  041132                          MOV     #5$,R2          ;POINTER TO CORRECT DATA
040752  004737  003672                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
040756  002642                                  MEMDAT                  ;POINTER TO DATA
040760  004737  004044                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
040764  002652                                  SPDAT                   ;POINTER TO DATA
040766                                          BGNSEG
040766  104404                                  TRAP    C$BSEG
040770  004737  004130                  1$:     JSR     PC,SETC         ;SET C BIT!
040774  042737  000017  041012                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
041002  050537  041012                          BIS     R5,2$           ;ADD ADDRESS TO INSTRUCTION
041006                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
041012  010000                          2$:     010000                  ;LOAD MAR
041014  042737  000017  041032                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
041022  050537  041032                          BIS     R5,3$           ;ADD ADDRESS TO INSTRUCTION
041026                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
041032  040560                          3$:     040400!<7*20>           ;BR   DEC A
041034                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
041040  061224                                  61224                   ;MOVE BR TO PORT4
041042  111237  002612                          MOVB    (R2),$GDDAT             ;PUT 'EXPECTED'' IN $GDDAT
041046  116104  000004                          MOVB    4(R1),R4        ;PUT 'FOUND IN R4
041052  123704  002612                          CMPB    $GDDAT,R4               ;DATA CORRECT?
041056  001413                                  BEQ     4$              ;BR IF YES
041060                                          ERROR   23              ;ALU ERROR
041076  104455                                  TRAP    C$ERDF
041100  000027                                  .WORD   23
041102  005475                                  .WORD   EM23
041104  010624                                  .WORD   ERR23
041106                                  4$:     ESCAPE  SEG
041106  104410                                  TRAP    C$ESCAPE
041110  000014                                  .WORD   10000$-.
041112  005202                                  INC     R2              ;NEXT DATA
041114  005205                                  INC     R5              ;NEXT ADDRESS
041116  022705  000010                          CMP     #10,R5          ;DONE YET?
041122  001322                                  BNE     1$              ;BR IF NO
041124                                          ENDSEG
041124                                  10000$:
041124  104405                                  TRAP    C$ESEG
041126                                          EXIT    TST
041126  104432                                  TRAP    C$EXIT
041130  000012                                  .WORD   L10171-.
041132    377     377     376           5$:     .BYTE   -1,-1,376,376,124,124,251,251
041135    376     124     124
041140    251     251
```

```
7880                                    .EVEN
7881  041142                            ENDTST
7882  041142                            L10171:
7883  041142  104401                            TRAP    C$ETST
7884
7885
```

```
7886
7887
7888
7889
7890
7891
```

```
                                .SBTTL  HARDWARE PARAMETER CODING SECTION



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

041144                                  BGNHRD
041144  000015                          .WORD L10172-L$HARD/2
041146                          L$HARD::

                                ;       GPRMD   WMP,0,0,7,0,7,YES
041146                                  GPRMA   ADDRES,2,0,160000,177776,YES
041146  001031                          .WORD   T$CODE
041150  041267                          .WORD   ADDRES
041152  160000                          .WORD   T$LOLIM
041154  177776                          .WORD   T$HILIM
041156                                  GPRMA   VECTOR,4,0,0,674,YES
041156  002031                          .WORD   T$CODE
041160  041306                          .WORD   VECTOR
041162  000000                          .WORD   T$LOLIM
041164  000674                          .WORD   T$HILIM
041166                                  GPRMD   PRIRTY,6,0,7000,4,7,YES
041166  003032                          .WORD   T$CODE
041170  041330                          .WORD   PRIRTY
041172  007000                          .WORD   7000
041174  000004                          .WORD   T$LOLIM
041176  000007                          .WORD   T$HILIM
                                ;       GPRMD   LNUNIT,10,0,3,0,3,YES
                                ;       GPRMD   SWPAC1,12,0,377,0,377,YES
                                ;       GPRMD   SWPAC2,14,0,377,0,377,YES
                                ;       GPRMD   LOOPBK,16,0,40000,0,1,YES

041200                                  ENDHRD
                                        .EVEN
041200                          L10172:

041200  044127  041511  020110  WMP:    .ASCIZ  'WHICH MICRO-PROCESSOR:''
041206  044515  051103  026517
041214  051120  041517  051505
041222  047523  035122     000
041227     060  046475  031070          .ASCIZ  ''0=M8200,4=M8204,6=M8206,7=M8207''
041234  030060  032054  046475
041242  031070  032060  033054
041250  046475  031070  033060
041256  033454  046475  031070
041264  033460     000
041267     103  051123  040440  ADDRES: .ASCIZ  /CSR ADDRESS : /
041274  042104  042522  051523
041302  035040  000040
```

```
 7948  041306  042526  052103  051117  VECTOR: .ASCIZ  /VECTOR ADDRESS : /
 7949  041314  040440  042104  042522
 7950  041322  051523  035040  000040
 7951  041330  051120  047511  044522  PRIRTY: .ASCIZ  /PRIORITY LEVEL : /
 7952  041336  054524  046040  053105
 7953  041344  046105  035040  000040
 7954  041352  044127  041511  020110  LNUNIT: .ASCIZ  /WHICH LINE UNIT (0-3)? 0=NONE,1=M8201,2=M8202,3=M8203 : /
 7955  041360  044514  042516  052440
 7956  041366  044516  020124  030050
 7957  041374  031455  037451  030040
 7958  041402  047075  047117  026105
 7959  041410  036461  034115  030062
 7960  041416  026061  036462  034115
 7961  041424  030062  026062  036463
 7962  041432  034115  030062  020063
 7963  041440  020072     000
 7964  041443     123  044527  041524  SWPAC1: .ASCIZ  /SWITCH PACK #1 (DDCMP LINE #) : /
 7965  041450  020110  040520  045503
 7966  041456  021440  020061  042050
 7967  041464  041504  050115  046040
 7968  041472  047111  020105  024443
 7969  041500  035040  000040
 7970  041504  053523  052111  044103  SWPAC2: .ASCIZ  /SWITCH PACK #2 (BM873 BOOT ADR) : /
 7971  041512  050040  041501  020113
 7972  041520  031043  024040  046502
 7973  041526  033470  020063  047502
 7974  041534  052117  040440  051104
 7975  041542  020051  020072     000
 7976  041547     127  046111  020114  LOOPBK: .ASCIZ  /WILL TEST CONNECTOR(S) BE USED ? 0=NO,1=YES : /
 7977  041554  042524  052123  041440
 7978  041562  047117  042516  052103
 7979  041570  051117  051450  020051
 7980  041576  042502  052440  042523
 7981  041604  020104  020077  036460
 7982  041612  047516  030454  054475
 7983  041620  051505  035040  000040
 7984
 7985                                  .EVEN
 7986
 7987
 7988
 7989
 7990
 7991
```

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

041626                                  BGNSFT
041626  000000                          .WORD L10173-L$SOFT/2
041630                          L$SOFT::


041630                                  ENDSFT
                                        .EVEN
041630                          L10173:

                                .EVEN





041630                                  ENDMOD


041630                          CORMAX:

041630  000000                          .WORD   0                       ;START OF NPR AREA (TEST 55)
        041732                  .=.+100
041732  000000                  MEMEND: .WORD   0                       ;END OF NPR AREA
041734                                  LASTAD
                                        .EVEN
041734  000000                          .WORD   0
041736  000000                          .WORD   0
041740                          L$LAST::
        000120                          LTN.ED=T$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  041267          7912    7945#
ADR   = 000020 G        1040#
ASSEMB= 000010           691
BIT0  = 000001 G        1013#   1318    1319    2928    5708
BIT00 = 000001 G        1002#   1013
BIT01 = 000002 G        1001#   1012
BIT02 = 000004 G        1000#   1011
BIT03 = 000010 G         999#   1010
BIT04 = 000020 G         998#   1009
BIT05 = 000040 G         997#   1008
BIT06 = 000100 G         996#   1007
BIT07 = 000200 G         995#   1006
BIT08 = 000400 G         994#   1005
BIT09 = 001000 G         993#   1004
BIT1  = 000002 G        1012#   1316    1318    1319
BIT10 = 002000 G         992#   1392    1400    3368    5504    5567
BIT11 = 004000 G         991#   3030    3048
BIT12 = 010000 G         990#   2665    3078    3096
BIT13 = 020000 G         989#   1395    1403    2676
BIT14 = 040000 G         988#
BIT15 = 100000 G         987#   2680
BIT2  = 000004 G        1011#   1319
BIT3  = 000010 G        1010#
BIT4  = 000020 G        1009#
BIT5  = 000040 G        1008#
BIT6  = 000100 G        1007#   1288    1289
BIT7  = 000200 G        1006#   1289
BIT8  = 000400 G        1005#   1374    3423    5506    5569
BIT9  = 001000 G        1004#   2979    2997    3368    3421    3423    5504    5567
BOE   = 000400 G        1044#
CLKX    002542          1102#
CLRALL  003326          1322#
CLRC    004112          1508#   5832    5896    5960    6024    6088    6152    6216    6280    6344    6410    6475    6541
                        6606    6671    6736    6802
CORMAX  041630          5704    5742    8022#
CRAMA   003554          1401    1410#
CZKMB   002000 G         697#
C$AU  = 000052           691#   2769
C$AUTO= 000061           691#   2560
C$BRK = 000022           691#
C$BSEG= 000004           691#   2867    2893    2930    2977    2996    3028    3047    3076    3095    3125    3177    3200
                        3241    3264    3305    3330    3367    3387    3458    3485    3531    3558    3606    3633    3678
                        3705    3750    3777    3822    3849    3894    3923    3970    4000    4048    4075    4120    4147
                        4192    4219    4264    4291    4336    4363    4408    4435    4480    4507    4552    4579    4624
                        4661    4899    4927    4992    5024    5083    5118    5776    5831    5895    5959    6023    6087
                        6151    6215    6279    6343    6409    6474    6540    6605    6670    6735    6801    6866    6932
                        6997    7062    7127    7192    7257    7322    7387    7452    7518    7583    7648    7713    7777
                        7842
C$BSUB= 000002           691#   4990
C$CEFG= 000045           691#
C$CLCK= 000062           691#
C$CLEA= 000012           691#   2731
C$CLOS= 000035           691#
C$CLP1= 000006           691#   4960
C$CVEC= 000036           691#
C$DCLN= 000044           691#
```

```
C$DODU= 000051   691#
C$DRPT= 000024   691#
C$DU  = 000053   691#  2751
C$EDIT= 000003   691#   769
C$ERDF= 000055   691#  1479  2805  2832  2875  2899  2938  2950  2985  3002  3036  3053  3084
                3101  3134  3146  3185  3209  3249  3274  3313  3336  3375  3393  3430  3470
                3498  3546  3574  3618  3646  3690  3718  3762  3790  3834  3862  3908  3938
                3985  4016  4060  4088  4132  4160  4204  4232  4276  4304  4348  4376  4420
                4448  4492  4520  4564  4592  4644  4672  4737  4749  4761  4856  4911  4940
                4955  5009  5043  5101  5131  5177  5183  5219  5225  5274  5320  5328  5372
                5419  5465  5531  5581  5625  5639  5681  5726  5791  5848  5912  5976  6040
                6104  6168  6232  6296  6360  6426  6491  6557  6622  6687  6752  6818  6883
                6949  7014  7079  7144  7209  7274  7339  7404  7469  7535  7600  7665  7730
                7794  7859
C$ERHR= 000056   691#
C$ERRO= 000060   691#
C$ERSF= 000054   691#
C$ERSO= 000057   691#
C$ESCA= 000010   691#  2797  2813  2880  2904  2943  2955  2990  3007  3041  3089  3139  3151
                3190  3214  3254  3279  3318  3341  3380  3398  3436  3475  3503  3579  3623
                3651  3695  3723  3767  3795  3839  3867  3913  3943  3990  4021  4065  4093
                4137  4165  4209  4237  4281  4309  4353  4381  4425  4453  4497  4525  4569
                4597  4649  4677  4861  4916  4945  5014  5048  5106  5136  5377  5424  5470
                5731  5796  5853  5917  5981  6045  6109  6173  6237  6301  6365  6431  6496
                6562  6627  6692  6757  6823  6888  6954  7019  7084  7149  7214  7279  7344
                7409  7474  7540  7605  7670  7735  7799  7864
C$ESEG= 000005   691#  2888  2912  2961  2994  3011  3045  3059  3093  3107  3157  3197  3222
                3261  3287  3328  3349  3384  3402  3482  3511  3555  3587  3630  3659  3702
                3731  3774  3803  3846  3875  3920  3951  3997  4029  4072  4101  4144  4173
                4216  4245  4288  4317  4360  4389  4432  4461  4504  4533  4576  4605  4657
                4686  4923  4969  5021  5056  5114  5144  5800  5861  5925  5989  6053  6117
                6181  6245  6309  6373  6439  6504  6570  6635  6700  6765  6831  6896  6962
                7027  7092  7157  7222  7287  7352  7417  7482  7548  7613  7678  7743  7807
                7872
C$ESUB= 000003   691#  5063
C$ETST= 000001   691#  2817  2839  2915  2964  3015  3063  3111  3160  3225  3290  3352  3405
                3441  3514  3590  3662  3734  3806  3878  3954  4032  4104  4176  4248  4320
                4392  4464  4536  4608  4689  4767  4883  4972  5066  5147  5191  5233  5282
                5337  5386  5433  5479  5538  5591  5647  5688  5754  5803  5872  5936  6000
                6064  6128  6192  6256  6320  6385  6450  6516  6581  6646  6711  6776  6842
                6907  6973  7038  7103  7168  7233  7298  7363  7428  7493  7559  7624  7689
                7754  7818  7883
C$EXIT= 000032   691#  5380  5427  5473  5747  5863  5927  5991  6055  6119  6183  6247  6311
                6375  6441  6506  6572  6637  6702  6767  6833  6898  6964  7029  7094  7159
                7224  7289  7354  7419  7484  7550  7615  7680  7745  7809  7874
C$GETB= 000026   691#
C$GETW= 000027   691#
C$GMAN= 000043   691#
C$GPHR= 000042   691#  2627
C$GPLO= 000030   691#
C$GPRI= 000040   691#
C$INIT= 000011   691#  2710
C$INLP= 000020   691#
C$MANI= 000050   691#
C$MEM = 000031   691#
C$MSG = 000023   691#  1830  1854  1878  1903  1928  1953  1978  2003  2027  2051  2068  2092
```

```
                    2117  2134  2151  2175  2192  2209  2234  2251  2275  2299  2324  2348  2373
                    2397  2414  2431  2455  2479  2503  2527
C$OPEN= 000034       691#
C$PNTB= 000014       691#  1814  1820  1827  1838  1844  1851  1862  1868  1875  1886  1892  1900
                    1911  1917  1925  1936  1942  1950  1961  1967  1975  1986  1992  2000  2011
                    2017  2024  2035  2041  2048  2059  2065  2076  2082  2089  2100  2106  2114
                    2125  2131  2142  2148  2159  2165  2172  2183  2189  2200  2206  2217  2223
                    2231  2242  2248  2259  2265  2272  2283  2289  2296  2307  2313  2321  2332
                    2338  2345  2356  2362  2370  2381  2387  2394  2405  2411  2422  2428  2439
                    2445  2452  2463  2469  2476  2487  2493  2500  2511  2517  2524  4872  4879
C$PNTF= 000017       691#
C$PNTS= 000016       691#
C$PNTX= 000015       691#
C$QIO = 000377       691#
C$RDBU= 000007       691#
C$REFG= 000047       691#  2590  2599  2605  2612
C$RESE= 000033       691#  2748
C$REVI= 000003       691#   768
C$RFLA= 000021       691#
C$RPT = 000025       691#  2547
C$SEFG= 000046       691#
C$SPRI= 000041       691#  5168  5174  5210  5216  5250  5267  5299  5317
C$SVEC= 000037       691#
C$TPRI= 000013       691#
DEBUG   003204      1298#  2827  2863  3302  3366  3419  3457  3530  3605  3677  3749  3821  3893
                    3969  4047  4119  4191  4263  4335  4407  4479  4551  4623  4703  4778  4898
                    4985  5082  5159  5203  5246  5295  5352  5398  5445  5493  5555  5604  5660
                    5701  5770  5823  5887  5951  6015  6079  6143  6207  6271  6335  6401  6466
                    6532  6597  6662  6727  6793  6858  6924  6989  7054  7119  7184  7249  7314
                    7379  7444  7510  7575  7640  7705  7769  7834
DFPTBL  002366 G     924#
DH0     006215      1748#  2061  2127  2144  2185  2202  2244  2407  2424
DH1     006216      1749#  1888  1913
DH2     006257      1755#  1816  1840  1864  2013  2037  2078  2161  2261  2285  2334  2383  2441
                    2465  2489  2513
DH27    006364      1769#  2309  2358
DH3     006273      1758#  1938  1963  1988
DH4     006333      1764#  2102  2219
DIAGMC= 000000        691
EF.CON= 000036 G     1020#  2604
EF.NEW= 000035 G     1021#  2598
EF.PWR= 000034 G     1022#
EF.RES= 000037 G     1019#  2611
EF.STA= 000040 G     1018#  2589
EM1     004530      1596#  2807
EM10    005071      1636#
EM11    005123      1641#  5374  5421  5467  5533  5583  5728
EM12    005134      1643#
EM13    005156      1646#  5627  5641  5683
EM14    005200      1649#
EM15    005223      1653#  5850  5914  5978  6042  6106  6170  6234  6298  6362  6428  6493  6559
                    6624  6689  6754  6820  7016
EM16    005266      1659#
EM17    005333      1667#
EM2     004556      1600#  2901  2952  3315  3338
EM20    005362      1671#
```

N 14

```
EM21    005427          1678#
EM22    005456          1682#
EM23    005475          1685#   3395    6885    6951    7081    7146    7211    7276    7341    7406    7471    7537    7602
                        7667    7732    7796    7861
EM24    005536          1691#
EM25    005722          1713#
EM26    005743          1716#   2834    3004    3038    3055    3086    3103    3377
EM27    005774          1721#   2877    2940    2987    3136    3148    3187    3211    3251    3276    3472    3500    3548
                        3576    3620    3648    3692    3720    3764    3792    3836    3864    3910    3940    3987    4018
                        4739    4751    4763    4858
EM28    006023          1725#   3432
EM29    006054          1730#   4062    4090    4134    4162    4206    4234    4278    4306    4350    4378    4422    4450
                        4494    4522    4566    4594    4646
EM3     004665          1612#   4913    4942    4957
EM30    004622          1606#   4674
EM31    005561          1695#   5179    5221    5322
EM32    005610          1699#   5185    5227    5330
EM33    005647          1705#   5276
EM34    005704          1710#   5793
EM35    005333          1666#
EM36    006107          1481    1735#
EM4     004713          1616#   5011    5045    5103
EM5     004741          1620#   5133
EM6     005002          1626#
EM7     005030          1630#
EM99    006164          1743#
ENDBUG  003276          1310#
ENDIT   013164          2607    2698    2702    2705    2707#
ERRFLG  002532          1098#
ERR1    006410  G       1808#   2808
ERR10   007402  G       1980#
ERR11   007514  G       2005#   5375    5422    5468    5534    5584    5729
ERR12   007622  G       2029#
ERR13   007730  G       2053#   5628    5642    5684
ERR14   010006  G       2070#
ERR15   010114  G       2094#   5851    5915    5979    6043    6107    6171    6235    6299    6363    6429    6494    6560
                        6625    6690    6755    6821    7017
ERR16   010226  G       2119#
ERR17   010304  G       2136#
ERR2    006516  G       1832#   2902    2953    3316    3339
ERR20   010362  G       2153#
ERR21   010470  G       2177#
ERR22   010546  G       2194#
ERR23   010624  G       2211#   3396    6886    6952    7082    7147    7212    7277    7342    7407    7472    7538    7603
                        7668    7733    7797    7862
ERR24   010736  G       2236#
ERR25   011014  G       2253#
ERR26   011122  G       2277#   2835    3005    3039    3056    3087    3104    3378
ERR27   011230  G       2301#   2878    2941    2988    3137    3149    3188    3212    3252    3277    3473    3501    3549
                        3577    3621    3649    3693    3721    3765    3793    3837    3865    3911    3941    3988    4019
                        4740    4752    4764    4859
ERR28   011342  G       2326#   3433
ERR29   011450  G       2350#   4063    4091    4135    4163    4207    4235    4279    4307    4351    4379    4423    4451
                        4495    4523    4567    4595    4647
ERR3    006624  G       1856#   4914    4943    4958
ERR30   011562  G       2375#   4675
```

```
ERR31   011670 G        2399#  5180   5222   5323
ERR32   011746 G        2416#  5186   5228   5331
ERR33   012024 G        2433#  5277
ERR34   012132 G        2457#  5794
ERR35   012240 G        2481#
ERR36   012346 G        1482   2505#
ERR4    006732 G        1880#  5012   5046   5104
ERR5    007044 G        1905#  5134
ERR6    007156 G        1930#
ERR7    007270 G        1955#
EVL   = 000004 G        1038#
E$END = 002100           691#
E$LOAD= 000035           691#   792
FLAG    002572          1115#
FMSG    004407          1582#  4869
FMX     004217          1554#  1811   1835   1859   1883   1908   1933   1958   1983   2008   2032   2056   2073
                        2097   2122   2139   2156   2180   2197   2214   2239   2256   2280   2304   2329   2353
                        2378   2402   2419   2436   2460   2484   2508
FM1     004210          1552#  1817   1841   1865   1889   1914   1939   1964   1989   2014   2038   2062   2079
                        2103   2128   2145   2162   2186   2203   2220   2245   2262   2286   2310   2335   2359
                        2384   2408   2425   2442   2466   2490   2514
FTIME   002622          1127#  2578   2584*
F$AU  = 000015           691#  2766   2768
F$AUTO= 000020           691#  2557   2559
F$BGN = 000040           691#   697   1808   1832   1856   1880   1905   1930   1955   1980   2005   2029   2053
                        2070   2094   2119   2136   2153   2177   2194   2211   2236   2253   2277   2301   2326
                        2350   2375   2399   2416   2433   2457   2481   2505   2537   2543   2550   2557   2572
                        2726   2745   2766   2789   2797   2813   2816   2825   2838   2861   2867   2880   2893
                        2904   2914   2925   2930   2943   2955   2963   2974   2977   2990   2996   3007   3014
                        3025   3028   3041   3047   3062   3073   3076   3089   3095   3110   3120   3125   3139
                        3151   3159   3171   3177   3190   3200   3214   3224   3235   3241   3254   3264   3279
                        3289   3300   3305   3318   3330   3341   3351   3362   3367   3380   3387   3398   3404
                        3417   3436   3440   3451   3458   3475   3485   3503   3513   3524   3531   3558   3579
                        3589   3600   3606   3623   3633   3651   3661   3672   3678   3695   3705   3723   3733
                        3744   3750   3767   3777   3795   3805   3816   3822   3839   3849   3867   3877   3888
                        3894   3913   3923   3943   3953   3964   3970   3990   4000   4021   4031   4042   4048
                        4065   4075   4093   4103   4114   4120   4137   4147   4165   4175   4186   4192   4209
                        4219   4237   4247   4258   4264   4281   4291   4309   4319   4330   4336   4353   4363
                        4381   4391   4402   4408   4425  -4435   4453   4463   4474   4480   4497   4507   4525
                        4535   4546   4552   4569   4579   4597   4607   4618   4624   4649   4661   4677   4688
                        4700   4766   4775   4861   4882   4893   4899   4916   4927   4945   4971   4983   4989
                        4992   5014   5024   5048   5062   5065   5077   5083   5106   5118   5136   5146   5157
                        5190   5201   5232   5244   5281   5293   5336   5347   5377   5380   5385   5396   5424
                        5427   5432   5443   5470   5473   5478   5490   5537   5553   5590   5602   5646   5658
                        5687   5699   5731   5747   5753   5768   5776   5796   5802   5821   5831   5853   5863
                        5871   5885   5895   5917   5927   5935   5949   5959   5981   5991   5999   6013   6023
                        6045   6055   6063   6077   6087   6109   6119   6127   6141   6151   6173   6183   6191
                        6205   6215   6237   6247   6255   6269   6279   6301   6311   6319   6333   6343   6365
                        6375   6384   6399   6409   6431   6441   6449   6464   6474   6496   6506   6515   6530
                        6540   6562   6572   6580   6595   6605   6627   6637   6645   6660   6670   6692   6702
                        6710   6725   6735   6757   6767   6775   6791   6801   6823   6833   6841   6856   6866
                        6888   6898   6906   6922   6932   6954   6964   6972   6987   6997   7019   7029   7037
                        7052   7062   7084   7094   7102   7117   7127   7149   7159   7167   7182   7192   7214
                        7224   7232   7247   7257   7279   7289   7297   7312   7322   7344   7354   7362   7377
                        7387   7409   7419   7427   7442   7452   7474   7484   7492   7508   7518   7540   7550
                        7558   7573   7583   7605   7615   7623   7638   7648   7670   7680   7688   7703   7713
```

F$CLEA= 000007
F$DU = 000016
F$END = 000041

7735 7745 7753 7767 7777 7799 7809 7817 7832 7842 7864 7874 7882
7906 8005 8020
691# 2726 2730
691# 2745 2750
691# 697 1831 1855 1879 1904 1929 1954 1979 2004 2028 2052 2069
2093 2118 2135 2152 2176 2193 2210 2235 2252 2276 2300 2325 2349
2374 2398 2415 2432 2456 2480 2504 2528 2541 2548 2561 2711 2732
2752 2770 2789 2797 2813 2816 2818 2825 2838 2840 2861 2880 2889
2904 2913 2914 2916 2925 2943 2955 2962 2963 2965 2974 2990 2995
3007 3012 3014 3016 3025 3041 3046 3060 3062 3064 3073 3089 3094
3108 3110 3112 3120 3139 3151 3158 3159 3161 3171 3190 3198 3214
3223 3224 3226 3235 3254 3262 3279 3288 3289 3291 3300 3318 3329
3341 3350 3351 3353 3362 3380 3385 3398 3403 3404 3406 3417 3436
3440 3442 3451 3475 3483 3503 3512 3513 3515 3524 3556 3579 3588
3589 3591 3600 3623 3631 3651 3660 3661 3663 3672 3695 3703 3723
3732 3733 3735 3744 3767 3775 3795 3804 3805 3807 3816 3839 3847
3867 3876 3877 3879 3888 3913 3921 3943 3952 3953 3955 3964 3990
3998 4021 4030 4031 4033 4042 4065 4073 4093 4102 4103 4105 4114
4137 4145 4165 4174 4175 4177 4186 4209 4217 4237 4246 4247 4249
4258 4281 4289 4309 4318 4319 4321 4330 4353 4361 4381 4390 4391
4393 4402 4425 4433 4453 4462 4463 4465 4474 4497 4505 4525 4534
4535 4537 4546 4569 4577 4597 4606 4607 4609 4618 4649 4658 4677
4687 4688 4690 4700 4766 4768 4775 4861 4882 4884 4893 4916 4924
4945 4970 4971 4973 4983 4989 5014 5022 5048 5057 5062 5064 5065
5067 5077 5106 5115 5136 5145 5146 5148 5157 5190 5192 5201 5232
5234 5244 5281 5283 5293 5336 5338 5347 5377 5380 5385 5387 5396
5424 5427 5432 5434 5443 5470 5473 5478 5480 5490 5537 5539 5553
5590 5592 5602 5646 5648 5658 5687 5689 5699 5731 5747 5753 5755
5768 5796 5801 5802 5804 5821 5853 5862 5863 5871 5873 5885 5917
5926 5927 5935 5937 5949 5981 5990 5991 5999 6001 6013 6045 6054
6055 6063 6065 6077 6109 6118 6119 6127 6129 6141 6173 6182 6183
6191 6193 6205 6237 6246 6247 6255 6257 6269 6301 6310 6311 6319
6321 6333 6365 6374 6375 6384 6386 6399 6431 6440 6441 6449 6451
6464 6496 6505 6506 6515 6517 6530 6562 6571 6572 6580 6582 6595
6627 6636 6637 6645 6647 6660 6692 6701 6702 6710 6712 6725 6757
6766 6767 6775 6777 6791 6823 6832 6833 6841 6843 6856 6888 6897
6898 6906 6908 6922 6954 6963 6964 6972 6974 6987 7019 7028 7029
7037 7039 7052 7084 7093 7094 7102 7104 7117 7149 7158 7159 7167
7169 7182 7214 7223 7224 7232 7234 7247 7279 7288 7289 7297 7299
7312 7344 7353 7354 7362 7364 7377 7409 7418 7419 7427 7429 7442
7474 7483 7484 7492 7494 7508 7540 7549 7550 7558 7560 7573 7605
7614 7615 7623 7625 7638 7670 7679 7680 7688 7690 7703 7735 7744
7745 7753 7755 7767 7799 7808 7809 7817 7819 7832 7864 7873 7874
7882 7884 7934 8012 8020

F$HARD= 000004
F$HW = 000013
F$INIT= 000006
F$JMP = 000050

691# 7906 7932
691# 922 937
691# 2572 2709
691# 2541 5380 5427 5473 5747 5863 5927 599, 6055 6119 6183 6247
6311 6375 6441 6506 6572 6637 6702 6767 6833 6898 6964 7029 7094
7159 7224 7289 7354 7419 7484 7550 7615 7680 7745 7809 7874

F$MOD = 000000
F$MSG = 000011

691# 697 8020
691# 1808 1829 1832 1853 1856 1877 1880 1902 1905 1927 1930 1952
1955 1977 1980 2002 2005 2026 2029 2050 2053 2067 2070 2091 2094
2116 2119 2133 2136 2150 2153 2174 2177 2191 2194 2208 2211 2233
2236 2250 2253 2274 2277 2298 2301 2323 2326 2347 2350 2372 2375
2396 2399 2413 2416 2430 2433 2454 2457 2478 2481 2502 2505 2526

```
F$PROT= 000021   691#  2550  2555
F$PWR = 000017   691#
F$RPT = 000012   691#  2537  2546
F$SEG = 000003   691#  2867  2887  2893  2911  2930  2960  2977  2993  2996  3010  3028  3044
                3047  3058  3076  3092  3095  3106  3125  3156  3177  3196  3200  3221  3241
                3260  3264  3286  3305  3327  3330  3348  3367  3383  3387  3401  3458  3481
                3485  3510  3531  3554  3558  3586  3606  3629  3633  3658  3678  3701  3705
                3730  3750  3773  3777  3802  3822  3845  3849  3874  3894  3919  3923  3950
                3970  3996  4000  4028  4048  4071  4075  4100  4120  4143  4147  4172  4192
                4215  4219  4244  4264  4287  4291  4316  4336  4359  4363  4388  4408  4431
                4435  4460  4480  4503  4507  4532  4552  4575  4579  4604  4624  4656  4661
                4685  4899  4922  4927  4968  4992  5020  5024  5055  5083  5113  5118  5143
                5776  5799  5831  5860  5895  5924  5959  5988  6023  6052  6087  6116  6151
                6180  6215  6244  6279  6308  6343  6372  6409  6438  6474  6503  6540  6569
                6605  6634  6670  6699  6735  6764  6801  6830  6866  6895  6932  6961  6997
                7026  7062  7091  7127  7156  7192  7221  7257  7286  7322  7351  7387  7416
                7452  7481  7518  7547  7583  7612  7648  7677  7713  7742  7777  7806  7842
                7871
F$SOFT= 000005   691#  8005  8010
F$SRV = 000010   691#
F$SUB = 000002   691#  4990  5062
F$SW  = 000014   691#   951   957
F$TEST= 000001   691#  2790  2816  2826  2838  2862  2914  2926  2963  2975  3014  3026  3062
                3074  3110  3121  3159  3172  3224  3236  3289  3301  3351  3363  3404  3418
                3440  3452  3513  3525  3589  3601  3661  3673  3733  3745  3805  3817  3877
                3889  3953  3965  4031  4043  4103  4115  4175  4187  4247  4259  4319  4331
                4391  4403  4463  4475  4535  4547  4607  4619  4688  4701  4766  4776  4882
                4894  4971  4984  5065  5078  5146  5158  5190  5202  5232  5245  5281  5294
                5336  5348  5385  5397  5432  5444  5478  5491  5537  5554  5590  5603  5646
                5659  5687  5700  5753  5769  5802  5822  5871  5886  5935  5950  5999  6014
                6063  6078  6127  6142  6191  6206  6255  6270  6319  6334  6384  6400  6449
                6465  6515  6531  6580  6596  6645  6661  6710  6726  6775  6792  6841  6857
                6906  6923  6972  6988  7037  7053  7102  7118  7167  7183  7232  7248  7297
                7313  7362  7378  7427  7443  7492  7509  7558  7574  7623  7639  7688  7704
                7753  7768  7817  7833  7882
GETPRM  012626   2614  2623#  2630
G$CNTO= 000200   691#
G$DELM= 000372   691#  1531  4783  4796  4805  4814  4823  4832  4841
G$DISP= 000003   691#
G$EXCP= 000400   691#
G$HILI= 000002   691#
G$LOLI= 000001   691#
G$NO  = 000000   691#
G$OFFS= 000400   691#  7911  7916  7921
G$OFSI= 000376   691#  7911  7916  7921
G$PRMA= 000001   691#  7911  7916
G$PRMD= 000002   691#  7921
G$PRML= 000000   691#
G$RADA= 000140   691#
G$RADB= 000000   691#
G$RADD= 000040   691#
G$RADL= 000120   691#
G$RADO= 000020   691#  7911  7916  7921
G$XFER= 000004   691#
G$YES = 000010   691#  7911  7916  7921
HELP  = 000000   682#   722   810   908  1235  2539  2544
```

```
HOE    = 100000 G          1051#
IBE    = 010000 G          1048#
IDU    = 000040 G          1041#
IER    = 020000 G          1049#
INIFLG  002662             1150#
INSTU   003570             1402    1413#
ISR    = 000100 G          1042#
IXE    = 004000 G          1047#
I$AU   = 000041             691#    2766#   2770#
I$AUTO= 000041              691#    2557#   2561#
I$CLN  = 000041             691#    2726#   2732#
I$DU   = 000041             691#    2745#   2752#
I$HRD  = 000041            7906#    7934#
I$INIT= 000041              691#    2572#   2711#
I$MOD  = 000041             691#     697#   8020#
I$MSG  = 000041             691#    1808#   1831#   1832#   1855#   1856#   1879#   1880#   1904#   1905#   1929#   1930#   1954#
                           1955#    1979#   1980#   2004#   2005#   2028#   2029#   2052#   2053#   2069#   2070#   2093#   2094#
                           2118#    2119#   2135#   2136#   2152#   2153#   2176#   2177#   2193#   2194#   2210#   2211#   2235#
                           2236#    2252#   2253#   2276#   2277#   2300#   2301#   2325#   2326#   2349#   2350#   2374#   2375#
                           2398#    2399#   2415#   2416#   2432#   2433#   2456#   2457#   2480#   2481#   2504#   2505#   2528#
I$PROT= 000040              691#    2550#
I$PTAB= 000041              691#
I$PWR  = 000041             691#
I$RPT  = 000041             691#    2537#   2548#
I$SEG  = 000041             691#    2789    2825    2861    2867#   2880    2889#   2893#   2904    2913#   2925    2930#   2943
                           2955    2962#   2974    2977#   2990    2995#   2996#   3007    3012#   3025    3028#   3041    3046#
                           3047#   3060#   3073    3076#   3089    3094#   3095#   3108#   3120    3125#   3139    3151    3158#
                           3171    3177#   3190    3198#   3200#   3214    3223#   3235    3241#   3254    3262#   3264#   3279
                           3288#   3300    3305#   3318    3329#   3330#   3341    3350#   3362    3367#   3380    3385#   3387#
                           3398    3403#   3417    3451    3458#   3475    3483#   3485#   3503    3512#   3524    3531#   3556#
                           3558#   3579    3588#   3600    3606#   3623    3631#   3633#   3651    3660#   3672    3678#   3695
                           3703#   3705#   3723    3732#   3744    3750#   3767    3775#   3777#   3795    3804#   3816    3822#
                           3839    3847#   3849#   3867    3876#   3888    3894#   3913    3921#   3923#   3943    3952#   3964
                           3970#   3990    3998#   4000#   4021    4030#   4042    4048#   4065    4073#   4075#   4093    4102#
                           4114    4120#   4137    4145#   4147#   4165    4174#   4186    4192#   4209    4217#   4219#   4237
                           4246#   4258    4264#   4281    4289#   4291#   4309    4318#   4330    4336#   4353    4361#   4363#
                           4381    4390#   4402    4408#   4425    4433#   4435#   4453    4462#   4474    4480#   4497    4505#
                           4507#   4525    4534#   4546    4552#   4569    4577#   4579#   4597    4606#   4618    4624#   4649
                           4658#   4661#   4677    4687#   4700    4775    4893    4899#   4916    4924#   4927#   4945    4970#
                           4983    4989    4992#   5014    5022#   5024#   5057#   5077    5083#   5106    5115#   5118#   5136
                           5145#   5157    5201    5244    5293    5347    5396    5443    5490    5553    5602    5658    5699
                           5768    5776#   5796    5801#   5821    5831#   5853    5862#   5885    5895#   5917    5926#   5949
                           5959#   5981    5990#   6013    6023#   6045    6054#   6077    6087#   6109    6118#   6141    6151#
                           6173    6182#   6205    6215#   6237    6246#   6269    6279#   6301    6310#   6333    6343#   6365
                           6374#   6399    6409#   6431    6440#   6464    6474#   6496    6505#   6530    6540#   6562    6571#
                           6595    6605#   6627    6636#   6660    6670#   6692    6701#   6725    6735#   6757    6766#   6791
                           6801#   6823    6832#   6856    6866#   6888    6897#   6922    6932#   6954    6963#   6987    6997#
                           7019    7028#   7052    7062#   7084    7093#   7117    7127#   7149    7158#   7182    7192#   7214
                           7223#   7247    7257#   7279    7288#   7312    7322#   7344    7353#   7377    7387#   7409    7418#
                           7442    7452#   7474    7483#   7508    7518#   7540    7549#   7573    7583#   7605    7614#   7638
                           7648#   7670    7679#   7703    7713#   7735    7744#   7767    7777#   7709    7808#   7832    7842#
                           7864    7873#
I$SETU= 000041              691#
I$SFT  = 000041            8005#    8012#
I$SRV  = 000041             691#
I$SUB  = 000041             691#    2789    2825    2861    2925    2974    3025    3073    3120    3171    3235    3300    3362
```

```
                    3417  3451  3524  3600  3672  3744  3816  3888  3964  4042  4114  4186  4258
                    4330  4402  4474  4546  4618  4700  4775  4893  4983  4989# 5062# 5064# 5077
                    5157  5201  5244  5293  5347  5396  5443  5490  5553  5602  5658  5699  5768
                    5821  5885  5949  6013  6077  6141  6205  6269  6333  6399  6464  6530  6595
                    6660  6725  6791  6856  6922  6987  7052  7117  7182  7247  7312  7377  7442
                    7508  7573  7638  7703  7767  7832
I$TST = 000041        691# 2789# 2797  2813  2816# 2818# 2825# 2838# 2840# 2861# 2914# 2916# 2925#
                    2963# 2965# 2974# 3014# 3016# 3025# 3062# 3064# 3073# 3110# 3112# 3120# 3159#
                    3161# 3171# 3224# 3226# 3235# 3289# 3291# 3300# 3351# 3353# 3362# 3404# 3406#
                    3417# 3436  3440# 3442# 3451# 3513# 3515# 3524# 3589# 3591# 3600# 3661# 3663#
                    3672# 3733# 3735# 3744# 3805# 3807# 3816# 3877# 3879# 3888# 3953# 3955# 3964#
                    4031# 4033# 4042# 4103# 4105# 4114# 4175# 4177# 4186# 4247# 4249# 4258# 4319#
                    4321# 4330# 4391# 4393# 4402# 4463# 4465# 4474# 4535# 4537# 4546# 4607# 4609#
                    4618# 4688# 4690# 4700# 4766# 4768# 4775# 4861  4882# 4884# 4893# 4971# 4973#
                    4983# 4989  5048  5065# 5067# 5077# 5146# 5148# 5157# 5190# 5192# 5201# 5232#
                    5234# 5244# 5281# 5283# 5293# 5336# 5338# 5347# 5377  5380  5385# 5387# 5396#
                    5424  5427  5432# 5434# 5443# 5470  5473  5478# 5480# 5490# 5537# 5539# 5553#
                    5590# 5592# 5602# 5646# 5648# 5658# 5687# 5689# 5699# 5731  5747  5753# 5755#
                    5768# 5802# 5804# 5821# 5863  5871# 5873# 5885# 5927  5935# 5937# 5949# 5991
                    5999# 6001# 6013# 6055  6063# 6065# 6077# 6119  6127# 6129# 6141# 6183  6191#
                    6193# 6205# 6247  6255# 6257# 6269# 6311  6319# 6321# 6333# 6375  6384# 6386#
                    6399# 6441  6449# 6451# 6464# 6506  6515# 6517# 6530# 6572  6580# 6582# 6595#
                    6637  6645# 6647# 6660# 6702  6710# 6712# 6725# 6767  6775# 6777# 6791# 6833
                    6841# 6843# 6856# 6898  6906# 6908# 6922# 6964  6972# 6974# 6987# 7029  7037#
                    7039# 7052# 7094  7102# 7104# 7117# 7159  7167# 7169# 7182# 7224  7232# 7234#
                    7247# 7289  7297# 7299# 7312# 7354  7362# 7364# 7377# 7419  7427# 7429# 7442#
                    7484  7492# 7494# 7508# 7550  7558# 7560# 7573# 7615  7623# 7625# 7638# 7680
                    7688# 7690# 7703# 7745  7753# 7755# 7767# 7809  7817# 7819# 7832# 7874  7882#
                    7884#
J$JMP = 000167        691# 2541
KMACTV  002562       1111#
KMCSR   002704       1186# 1300  1809  1833  1857  1881  1906  1931  1956  1981  2006  2030  2054
                    2071  2095  2120  2137  2154  2178  2195  2212  2237  2254  2278  2302  2327
                    2351  2376  2400  2417  2434  2458  2482  2506  2634* 2790  2889  2931  2078
                    3029  3077  3324
KMCSRH  002706       1187# 1287* 1288* 1289* 1316* 1318* 1319* 2636* 2637*
KMCTL   002710       1188# 2639* 2640* 3127
KMNUM   002564       1112#
KMPO4   002712       1189# 1461* 1475  1497* 2642* 2643* 3174
KMPO6   002714       1190# 1317* 2645* 2646* 3238  5493  5494
KMRLVL  002676       1183# 1425* 2650* 2651*
KMRVEC  002674       1182# 1423* 2648*
KMTLVL  002702       1185# 1426* 2656* 2657*
KMTVEC  002700       1184# 1424* 2653* 2654*
LNUNIT  041352       7954#
LOCK    002414       1081# 2891*
LOE   = 040000 G     1050#
LOGDEV  002524       1095# 1810  1834  1858  1882  1907  1932  1957  1982  2007  2031  2055  2072
                    2096  2121  2138  2155  2179  2196  2213  2238  2255  2279  2303  2328  2352
                    2377  2401  2418  2435  2459  2483  2507  2618* 2624* 2626
LOKFLG  002664       1152#
LOOPBK  041547       7976#
LOT   = 000010 G     1039#
LTN.ED= 000120       8032#
L$ACP   002110 G      799#
L$APT   002036 G      757#
```

L$AU	013174	G	784	2766#						
L$AUT	002070	G	783#							
L$AUTO	012470	G	800	2557#						
L$CCP	002106	G	797#							
L$CLEA	013166	G	798	2726#						
L$CO	002032	G	753#							
L$DEPO	002011	G	735#							
L$DESC	003124	G	790	1222#						
L$DESP	002076	G	789#							
L$DEVP	002060	G	775#							
L$DISP	002124	G	760	825#						
L$DLY	002116	G	805#	1533	4785	4798	4807	4816	4825	4834	4843
L$DTP	002040	G	759#							
L$DTYP	002034	G	755#							
L$DU	013170	G	786	2745#						
L$DUT	002072	G	785#							
L$DVTY	003116	G	776	1218#						
L$EF	002052	G	770#							
L$ENVI	002044	G	763#							
L$ETP	002102	G	793#							
L$EXP1	002046	G	765#							
L$EXP4	002064	G	779#							
L$EXP5	002066	G	781#							
L$HARD	041146	G	742	7906	7907#					
L$HIME	002120	G	807#							
L$HPCP	002016	G	741#							
L$HPTP	002022	G	745#							
L$HW	002366	G	746	922	923#					
L$ICP	002104	G	795#							
L$INIT	012472	G	796	2572#						
L$LADP	002026	G	749#							
L$LAST	041740	G	750	8031#						
L$LOAD	002100	G	791#							
L$LUN	002074	G	787#							
L$MREV	002050	G	767#							
L$NAME	002000	G	724#							
L$PRIO	002042	G	761#							
L$PROT	012462	G	802	2550#						
L$PRT	002112	G	801#							
L$REPP	002062	G	777#							
L$REV	002010	G	733#							
L$RPT	012454	G	778	2537#						
L$SOFT	041630	G	8005	8006#						
L$SPC	002056	G	773#							
L$SPCP	002020	G	743#							
L$SPTP	002024	G	747#							
L$STA	002030	G	751#							
L$SW	002412	G	748	951	952#					
L$TEST	002114	G	803#							
L$TIML	002014	G	739#							
L$UNIT	002012	G	737#							
L10000	002410		922	937#						
L10001	002412		951	957#						
L10002	006514		1829#							
L10003	006622		1853#							
L10004	006730		1877#							

```
L10005  007042          1902#
L10006  007154          1927#
L10007  007266          1952#
L10010  007400          1977#
L10011  007512          2002#
L10012  007620          2026#
L10013  007726          2050#
L10014  010004          2067#
L10015  010112          2091#
L10016  010224          2116#
L10017  010302          2133#
L10020  010360          2150#
L10021  010466          2174#
L10022  010544          2191#
L10023  010622          2208#
L10024  010734          2233#
L10025  011012          2250#
L10026  011120          2274#
L10027  011226          2298#
L10030  011340          2323#
L10031  011446          2347#
L10032  011560          2372#
L10033  011666          2396#
L10034  011744          2413#
L10035  012022          2430#
L10036  012130          2454#
L10037  012236          2478#
L10040  012344          2502#
L10041  012452          2526#
L10042  012460          2542    2546#
L10044  012470          2559#
L10045  013164          2709#
L10046  013166          2730#
L10047  013172          2750#
L10050  013174          2768#
L10051  013316          2798    2814    2816#
L10052  013364          2838#
L10053  013574          2914#
L10054  013744          2963#
L10055  014110          3014#
L10056  014250          3062#
L10057  014410          3110#
L10060  014562          3159#
L10061  014766          3224#
L10062  015172          3289#
L10063  015374          3351#
L10064  015560          3404#
L10065  015672          3437    3440#
L10066  016132          3513#
L10067  016426          3589#
L10070  016666          3661#
L10071  017126          3733#
L10072  017366          3805#
L10073  017626          3877#
L10074  020116          3953#
L10075  020420          4031#
```

```
L10076  020660          4103#
L10077  021120          4175#
L10100  021360          4247#
L10101  021620          4319#
L10102  022060          4391#
L10103  022320          4463#
L10104  022560          4535#
L10105  023020          4607#
L10106  023326          4688#
L10107  023724          4766#
L10110  024410          4862    4882#
L10111  024720          4971#
L10112  025260          5049    5065#
L10113  025256          5062#
L10114  025566          5146#
L10115  025734          5190#
L10116  026076          5232#
L10117  026250          5281#
L10120  026454          5336#
L10121  026630          5378    5381    5385#
L10122  027004          5425    5428    5432#
L10123  027154          5471    5474    5478#
L10124  027346          5537#
L10125  027566          5590#
L10126  030014          5646#
L10127  030152          5687#
L10130  030374          5732    5748    5753#
L10131  030530          5802#
L10132  030750          5864    5871#
L10133  031160          5928    5935#
L10134  031370          5992    5999#
L10135  031600          6056    6063#
L10136  032010          6120    6127#
L10137  032220          6184    6191#
L10140  032430          6248    6255#
L10141  032640          6312    6319#
L10142  033052          6376    6384#
L10143  033262          6442    6449#
L10144  033472          6507    6515#
L10145  033702          6573    6580#
L10146  034112          6638    6645#
L10147  034322          6703    6710#
L10150  034532          6768    6775#
L10151  034742          6834    6841#
L10152  035152          6899    6906#
L10153  035362          6965    6972#
L10154  035572          7030    7037#
L10155  036002          7095    7102#
L10156  036212          7160    7167#
L10157  036422          7225    7232#
L10160  036632          7290    7297#
L10161  037042          7355    7362#
L10162  037252          7420    7427#
L10163  037462          7485    7492#
L10164  037672          7551    7558#
L10165  040102          7616    7623#
```

```
L10166  040312          7681    7688#
L10167  040522          7746    7753#
L10170  040732          7810    7817#
L10171  041142          7875    7882#
L10172  041200          7906    7933#
L10173  041630          8005    8011#
MASKX   002544          1103#
MDATA   002460          1088#
MEMDAT  002642          1139#   5827    5891    5955    6019    6083    6147    6211    6275    6339    6405    6470    6536
                        6601    6666    6731    6797    6862    6928    6993    7058    7123    7188    7253    7318    7383
                        7448    7514    7579    7644    7709    7773    7838
MEMEND  041732          1108    8026#
MEMLD   003672          1448#   5771    5826    5890    5954    6018    6082    6146    6210    6274    6338    6404    6469
                        6535    6600    6665    6730    6796    6861    6927    6992    7057    7122    7187    7252    7317
                        7382    7447    7513    7578    7643    7708    7772    7837
MEMLD2  003702          1455#
MEMLIM  002556          1108#   5740
MEMSET  003456          1383#
MEMSZ   002560          1109#   2695*   2703*
MRO     002576          1117#   1479    2110    2317    2366    2805    2832    2875    2899    2927*   2938    2950    2985
                        3002    3036    3053    3084    3101    3122*   3134    3146    3172*   3185    3209    3236*   3249
                        3274    3313    3336    3375    3393    3430    3453*   3470    3498    3526*   3546    3574    3602*
                        3618    3646    3674*   3690    3718    3746*   3762    3790    3818*   3834    3862    3890*   3908
                        3938    3966*   3985    4016    4044*   4060    4088    4116*   4132    4160    4188*   4204    4232
                        4260*   4276    4304    4332*   4348    4376    4404*   4420    4448    4476*   4492    4520    4548*
                        4564    4592    4644    4672    4729*   4737    4741*   4749    4753*   4761    4856    4911    4940
                        4955    5009    5043    5101    5131    5177    5183    5219    5225    5274    5320    5328    5372
                        5419    5465    5531    5581    5625    5639    5681    5726    5791    5848    5912    5976    6040
                        6104    6168    6232    6296    6360    6426    6491    6557    6622    6687    6752    6818    6883
                        6949    7014    7079    7144    7209    7274    7339    7404    7469    7535    7600    7665    7730
                        7794    7859
NEWST   012620          2594    2601    2616#
NEXT    002412          1080#
NPRSET  003626          1430#   5354    5400    5447    5495    5561    5605    5661    5711
ONE     002554          1107#
O$APTS= 000000          691#    751
O$AU  = 000001          691#    721#    783
O$BGNR= 000001          691#    721#    777
O$BGNS= 000000          691#    743
O$DU  = 000001          691#    721#    785
O$ERRT= 000000          691#    793
O$GNSW= 000001          691#    721#    747
O$POIN= 000001          691#    721#    809
O$SETU= 000000          691#    737     8029
PNT   = 001000 G        1045#
PONE    002604          1120#   2582*   2593*   4863    4865*
POP2SP= 022626          1060#
PRDEL   004146          1529#   5159    5348
PRI   = 002000 G        1046#
PRIRTY  041330          7922    7951#
PRI00 = 000000 G        1034#   5173    5215
PRI01 = 000040 G        1033#
PRI02 = 000100 G        1032#
PRI03 = 000140 G        1031#
PRI04 = 000200 G        1030#
PRI05 = 000240 G        1029#
```

```
PRI06 = 000300 G    1028#
PRI07 = 000340 G    1027#  5167   5209
PSTACK  002526      1096#  2577*
QV.FLG  002665      1153#
RAMDAT  003422      1367#
REGADR  002716      1194#
RETADR  002534      1099#
RUN     002574      1116#
SAVACT  002566      1113#
SAVE4   002624      1128#  2580*  2585   2810   5588
SAVE6   002626      1129#  2581*  2586   2811   5587
SAVNUM  002570      1114#
SAVPC   002550      1105#
SAVSP   002546      1104#
SCLK    002630      1130#  4778*  4853*  4866
SETBR0  003352      1332#
SETBR1  003362      1338#
SETBR4  003372      1345#
SETBR7  003402      1352#
SETC    004130      1518#  6867   6933   6998   7063   7128   7193   7258   7323   7388   7453   7519   7584
                    7649   7714   7778   7843
SETVEC  003604      1420#  5162   5204   5258   5308
SETZ    003412      1360#
SFPTBL  002412 G     953#
SIBS10  002632      1131#  4703*  4708*  4715   4730
SIBS11  002634      1132#  4704*  4711*  4742
SIBS12  002636      1133#
SIBS13  002640      1134#  4705*  4714*  4724   4754
SMSG    004457      1589#  4876
SPDAT   002652      1142#  5829   5893   5957   6021   6085   6149   6213   6277   6341   6407   6472   6538
                    6603   6668   6733   6799   6864   6930   6995   7060   7125   7190   7255   7320   7385
                    7450   7516   7581   7646   7711   7775   7840
SPLD    004044      1490#  5773   5828   5892   5956   6020   6084   6148   6212   6276   6340   6406   6471
                    6537   6602   6667   6732   6798   6863   6929   6994   7059   7124   7189   7254   7319
                    7384   7449   7515   7580   7645   7710   7774   7839
SSTACK  003116      1198#  2575
STAT    002540      1101#
STAT1   002666      1175#  2659*  2661*  2665*  2676*  2680*  2683*  5251   5300
STAT2   002670      1176#  2686*  2688*
STAT3   002672      1177#
STOP    027360      5556#
STRTSW  002536      1100#
SUBRPC  002530      1097#
SVCGBL= 000000       691#   697   704#   724    725    733    734    735    736    737    738    739    740
                     741    742    743    744    745    746    747    748    749    750    751    752    753
                     754    755    756    757    758    759    760    761    762    763    764    765    766
                     767    768    770    771    773    774    775    776    777    778    779    780    781
                     782    783    784    785    786    787    788    789    790    791    792    793    794
                     795    796    797    798    799    800    801    802    803    804    805    806    807
                     808    825    826    923    924    925    952    953    954   1218   1219   1222   1223
                    1808   1809   1832   1833   1856   1857   1880   1881   1905   1906   1930   1931   1955
                    1956   1980   1981   2005   2006   2029   2030   2053   2054   2070   2071   2094   2095
                    2119   2120   2136   2137   2153   2154   2177   2178   2194   2195   2211   2212   2236
                    2237   2253   2254   2277   2278   2301   2302   2326   2327   2350   2351   2375   2376
                    2399   2400   2416   2417   2433   2434   2457   2458   2481   2482   2505   2506   2537
                    2538   2550   2551   2557   2558   2572   2573   2726   2727   2745   2746   2766   2767
```

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0193

SVCINS= 000000

7907 7908 8006 8007 8031# 8032
691# 701# 725 726 727 728 729 730 731 732 733 734 735
736 737 738 739 740 741 742 743 744 745 746 747 748
749 750 751 752 753 754 755 756 757 758 759 760 761
762 763 764 765 766 767 768 769 770 771 772 773 774
775 776 777 778 779 780 781 782 783 784 785 786 787
788 789 790 791 792 793 794 795 796 797 798 799 800
801 802 803 804 805 806 807 808 809 824 825 826 827
828 829 830 831 832 833 834 835 836 837 838 839 840
841 842 843 844 845 846 847 848 849 850 851 852 853
854 855 856 857 858 859 860 861 862 863 864 865 866
867 868 869 870 871 872 873 874 875 876 877 878 879
880 881 882 883 884 885 886 887 888 889 890 891 892
893 894 895 896 897 898 899 900 901 902 903 904 905
906 922 923 951 952 1219 1220 1221 1223 1228 1229 1479 1480
1481 1482 1483 1531 1532 1533 1534 1535 1536 1537 1538 1539 1809
1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822
1823 1824 1825 1826 1827 1828 1829 1830 1831 1833 1834 1835 1836
1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849
1850 1851 1852 1853 1854 1855 1857 1858 1859 1860 1861 1862 1863
1864 1865 1866 1867 1868 1869 1870 1871 1872 1873 1874 1875 1876
1877 1878 1879 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890
1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1903
1904 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917
1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1931
1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944
1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1956 1957 1958
1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971
1972 1973 1974 1975 1976 1977 1978 1979 1981 1982 1983 1984 1985
1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998
1999 2000 2001 2002 2003 2004 2006 2007 2008 2009 2010 2011 2012
2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025
2026 2027 2028 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039
2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052
2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066
2067 2068 2069 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080
2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093
2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107
2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2120 2121
2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134
2135 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148
2149 2150 2151 2152 2154 2155 2156 2157 2158 2159 2160 2161 2162
2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175
2176 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189
2190 2191 2192 2193 2195 2196 2197 2198 2199 2200 2201 2202 2203
2204 2205 2206 2207 2208 2209 2210 2212 2213 2214 2215 2216 2217
2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230
2231 2232 2233 2234 2235 2237 2238 2239 2240 2241 2242 2243 2244
2245 2246 2247 2248 2249 2250 2251 2252 2254 2255 2256 2257 2258
2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271
2272 2273 2274 2275 2276 2278 2279 2280 2281 2282 2283 2284 2285
2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298
2299 2300 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312
2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325
2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339
2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2351 2352 2353

CROSS REFERENCE TABLE -- USER SYMBOLS

2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366
2367	2368	2369	2370	2371	2372	2373	2374	2376	2377	2378	2379	2380
2381	2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393
2394	2395	2396	2397	2398	2400	2401	2402	2403	2404	2405	2406	2407
2408	2409	2410	2411	2412	2413	2414	2415	2417	2418	2419	2420	2421
2422	2423	2424	2425	2426	2427	2428	2429	2430	2431	2432	2434	2435
2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447	2448
2449	2450	2451	2452	2453	2454	2455	2456	2458	2459	2460	2461	2462
2463	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475
2476	2477	2478	2479	2480	2482	2483	2484	2485	2486	2487	2488	2489
2490	2491	2492	2493	2494	2495	2496	2497	2498	2499	2500	2501	2502
2503	2504	2506	2507	2508	2509	2510	2511	2512	2513	2514	2515	2516
2517	2518	2519	2520	2521	2522	2523	2524	2525	2526	2527	2528	2541
2542	2543	2547	2548	2560	2561	2589	2590	2591	2592	2593	2598	2599
2600	2601	2602	2604	2605	2606	2607	2608	2611	2612	2613	2614	2615
2626	2627	2628	2629	2630	2631	2710	2711	2731	2732	2748	2749	2751
2752	2769	2770	2797	2798	2799	2805	2806	2807	2808	2809	2813	2814
2815	2817	2818	2832	2833	2834	2835	2836	2839	2840	2867	2868	2875
2876	2877	2878	2879	2880	2881	2882	2888	2889	2893	2894	2899	2900
2901	2902	2903	2904	2905	2906	2912	2913	2915	2916	2930	2931	2938
2939	2940	2941	2942	2943	2944	2945	2950	2951	2952	2953	2954	2955
2956	2957	2961	2962	2964	2965	2977	2978	2985	2986	2987	2988	2989
2990	2991	2992	2994	2995	2996	2997	3002	3003	3004	3005	3006	3007
3008	3009	3011	3012	3015	3016	3028	3029	3036	3037	3038	3039	3040
3041	3042	3043	3045	3046	3047	3048	3053	3054	3055	3056	3057	3059
3060	3063	3064	3076	3077	3084	3085	3086	3087	3088	3089	3090	3091
3093	3094	3095	3096	3101	3102	3103	3104	3105	3107	3108	3111	3112
3125	3126	3134	3135	3136	3137	3138	3139	3140	3141	3146	3147	3148
3149	3150	3151	3152	3153	3157	3158	3160	3161	3177	3178	3185	3186
3187	3188	3189	3190	3191	3192	3197	3198	3200	3201	3209	3210	3211
3212	3213	3214	3215	3216	3222	3223	3225	3226	3241	3242	3245	3250
3251	3252	3253	3254	3255	3256	3261	3262	3264	3265	3274	3275	3276
3277	3278	3279	3280	3281	3287	3288	3290	3291	3305	3306	3313	3314
3315	3316	3317	3318	3319	3320	3328	3329	3330	3331	3336	3337	3338
3339	3340	3341	3342	3343	3349	3350	3352	3353	3367	3368	3375	3376
3377	3378	3379	3380	3381	3382	3384	3385	3387	3388	3393	3394	3395
3396	3397	3398	3399	3400	3402	3403	3405	3406	3430	3431	3432	3433
3434	3436	3437	3438	3441	3442	3458	3459	3470	3471	3472	3473	3474
3475	3476	3477	3482	3483	3485	3486	3498	3499	3500	3501	3502	3503
3504	3505	3511	3512	3514	3515	3531	3532	3546	3547	3548	3549	3550
3555	3556	3558	3559	3574	3575	3576	3577	3578	3579	3580	3581	3587
3588	3590	3591	3606	3607	3618	3619	3620	3621	3622	3623	3624	3625
3630	3631	3633	3634	3646	3647	3648	3649	3650	3651	3652	3653	3659
3660	3662	3663	3678	3679	3690	3691	3692	3693	3694	3695	3696	3697
3702	3703	3705	3706	3718	3719	3720	3721	3722	3723	3724	3725	3731
3732	3734	3735	3750	3751	3762	3763	3764	3765	3766	3767	3768	3769
3774	3775	3777	3778	3790	3791	3792	3793	3794	3795	3796	3797	3803
3804	3806	3807	3822	3823	3834	3835	3836	3837	3838	3839	3840	3841
3846	3847	3849	3850	3862	3863	3864	3865	3866	3867	3868	3869	3875
3876	3878	3879	3894	3895	3908	3909	3910	3911	3912	3913	3914	3915
3920	3921	3923	3924	3938	3939	3940	3941	3942	3943	3944	3945	3951
3952	3954	3955	3970	3971	3985	3986	3987	3988	3989	3990	3991	3992
3997	3998	4000	4001	4016	4017	4018	4019	4020	4021	4022	4023	4029
4030	4032	4033	4048	4049	4060	4061	4062	4063	4064	4065	4066	4067
4072	4073	4075	4076	4088	4089	4090	4091	4092	4093	4094	4095	4101
4102	4104	4105	4120	4121	4132	4133	4134	4135	4136	4137	4138	4139

N 15

4144 4145 4147 4148 4160 4161 4162 4163 4164 4165 4166 4167 4173
4174 4176 4177 4192 4193 4204 4205 4206 4207 4208 4209 4210 4211
4216 4217 4219 4220 4232 4233 4234 4235 4236 4237 4238 4239 4245
4246 4248 4249 4264 4265 4276 4277 4278 4279 4280 4281 4282 4283
4288 4289 4291 4292 4304 4305 4306 4307 4308 4309 4310 4311 4317
4318 4320 4321 4336 4337 4348 4349 4350 4351 4352 4353 4354 4355
4360 4361 4363 4364 4376 4377 4378 4379 4380 4381 4382 4383 4389
4390 4392 4393 4408 4409 4420 4421 4422 4423 4424 4425 4426 4427
4432 4433 4435 4436 4448 4449 4450 4451 4452 4453 4454 4455 4461
4462 4464 4465 4480 4481 4492 4493 4494 4495 4496 4497 4498 4499
4504 4505 4507 4508 4520 452: 4522 4523 4524 4525 4526 4527 4533
4534 4536 4537 4552 4553 4564 4565 4566 4567 4568 4569 4570 4571
4576 4577 4579 4580 4592 4593 4594 4595 4596 4597 4598 4599 4605
4606 4608 4609 4624 4625 4644 4645 4646 4647 4648 4649 4650 4651
4657 4658 4661 4662 4672 4673 4674 4675 4676 4677 4678 4679 4686
4687 4689 4690 4737 4738 4739 4740 4741 4749 4750 4751 4752 4753
4761 4762 4763 4764 4765 4767 4768 4783 4784 4785 4786 4787 4788
4789 4790 4791 4796 4797 4798 4799 4800 4801 4802 4803 4804 4805
4806 4807 4808 4809 4810 4811 4812 4813 4814 4815 4816 4817 4818
4819 4820 4821 4822 4823 4824 4825 4826 4827 4828 4829 4830 4831
4832 4833 4834 4835 4836 4837 4838 4839 4840 4841 4842 4843 4844
4845 4846 4847 4848 4849 4856 4857 4858 4859 4860 4861 4862 4863
4869 4870 4871 4872 4873 4874 4876 4877 4878 4879 4880 4881 4883
4884 4899 4900 4911 4912 4913 4914 4915 4916 4917 4918 4923 4924
4927 4928 4940 4941 4942 4943 4944 4945 4946 4947 4955 4956 4957
4958 4959 4960 4961 4969 4970 4972 4973 4990 4991 4992 4993 5009
5010 5011 5012 5013 5014 5015 5016 5021 5022 5024 5025 5043 5044
5045 5046 5047 5048 5049 5050 5056 5057 5063 5064 5066 5067 5083
5084 5101 5102 5103 5104 5105 5106 5107 5108 5114 5115 5118 5119
5131 5132 5133 5134 5135 5136 5137 5138 5144 5145 5147 5148 5167
5168 5169 5173 5174 5175 5177 5178 5179 5180 5181 5183 5184 5185
5186 5187 5191 5192 5209 5210 5211 5215 5216 5217 5219 5220 5221
5222 5223 5225 5226 5227 5228 5229 5233 5234 5249 5250 5251 5266
5267 5268 5274 5275 5276 5277 5278 5282 5283 5298 5299 5300 5316
5317 5318 5320 5321 5322 5323 5324 5328 5329 5330 5331 5332 5337
5338 5372 5373 5374 5375 5376 5377 5378 5379 5380 5381 5382 5386
5387 5419 5420 5421 5422 5423 5424 5425 5426 5427 5428 5429 5433
5434 5465 5466 5467 5468 5469 5470 5471 5472 5473 5474 5475 5479
5480 5531 5532 5533 5534 5535 5538 5539 5581 5582 5583 5584 5585
5591 5592 5625 5626 5627 5628 5629 5639 5640 5641 5642 5643 5647
5648 5681 5682 5683 5684 5685 5688 5689 5726 5727 5728 5729 5730
5731 5732 5733 5747 5748 5749 5754 5755 5776 5777 5791 5792 5793
5794 5795 5796 5797 5798 5800 5801 5803 5804 5831 5832 5848 5849
5850 5851 5852 5853 5854 5855 5861 5862 5863 5864 5865 5872 5873
5895 5896 5912 5913 5914 5915 5916 5917 5918 5919 5925 5926 5927
5928 5929 5936 5937 5959 5960 5976 5977 5978 5979 5980 5981 5982
5983 5989 5990 5991 5992 5993 6000 6001 6023 6024 6040 6041 6042
6043 6044 6045 6046 6047 6053 6054 6055 6056 6057 6064 6065 6087
6088 6104 6105 6106 6107 6108 6109 6110 6111 6117 6118 6119 6120
6121 6128 6129 6151 6152 6168 6169 6170 6171 6172 6173 6174 6175
6181 6182 6183 6184 6185 6192 6193 6215 6216 6232 6233 6234 6235
6236 6237 6238 6239 6245 6246 6247 6248 6249 6256 6257 6279 6280
6296 6297 6298 6299 6300 6301 6302 6303 6309 6310 6311 6312 6313
6320 6321 6343 6344 6360 6361 6362 6363 6364 6365 6366 6367 6373
6374 6375 6376 6377 6385 6386 6409 6410 6426 6427 6428 6429 6430
6431 6432 6433 6439 6440 6441 6442 6443 6450 6451 6474 6475 6491

SVCSUB= 000000
SVCTAG= 000000

6492 6493 6494 6495 6496 6497 6498 6504 6505 6506 6507 6508 6516
6517 6540 6541 6557 6558 6559 6560 6561 6562 6563 6564 6570 6571
6572 6573 6574 6581 6582 6605 6606 6622 6623 6624 6625 6626 6627
6628 6629 6635 6636 6637 6638 6639 6646 6647 6670 6671 6687 6688
6689 6690 6691 6692 6693 6694 6700 6701 6702 6703 6704 6711 6712
6735 6736 6752 6753 6754 6755 6756 6757 6758 6759 6765 6766 6767
6768 6769 6776 6777 6801 6802 6818 6819 6820 6821 6822 6823 6824
6825 6831 6832 6833 6834 6835 6842 6843 6866 6867 6883 6884 6885
6886 6887 6888 6889 6890 6896 6897 6898 6899 6900 6907 6908 6932
6933 6949 6950 6951 6952 6953 6954 6955 6956 6962 6963 6964 6965
6966 6973 6974 6997 6998 7014 7015 7016 7017 7018 7019 7020 7021
7027 7028 7029 7030 7031 7038 7039 7062 7063 7079 7080 7081 7082
7083 7084 7085 7086 7092 7093 7094 7095 7096 7103 7104 7127 7128
7144 7145 7146 7147 7148 7149 7150 7151 7157 7158 7159 7160 7161
7168 7169 7192 7193 7209 7210 7211 7212 7213 7214 7215 7216 7222
7223 7224 7225 7226 7233 7234 7257 7258 7274 7275 7276 7277 7278
7279 7280 7281 7287 7288 7289 7290 7291 7298 7299 7322 7323 7339
7340 7341 7342 7343 7344 7345 7346 7352 7353 7354 7355 7356 7363
7364 7387 7388 7404 7405 7406 7407 7408 7409 7410 7411 7417 7418
7419 7420 7421 7428 7429 7452 7453 7469 7470 7471 7472 7473 7474
7475 7476 7482 7483 7484 7485 7486 7493 7494 7518 7519 7535 7536
7537 7538 7539 7540 7541 7542 7548 7549 7550 7551 7552 7559 7560
7583 7584 7600 7601 7602 7603 7604 7605 7606 7607 7613 7614 7615
7616 7617 7624 7625 7648 7649 7665 7666 7667 7668 7669 7670 7671
7672 7678 7679 7680 7681 7682 7689 7690 7713 7714 7730 7731 7732
7733 7734 7735 7736 7737 7743 7744 7745 7746 7747 7754 7755 7777
7778 7794 7795 7796 7797 7798 7799 7800 7801 7807 7808 7809 7810
7811 7818 7819 7842 7843 7859 7860 7861 7862 7863 7864 7865 7866
7872 7873 7874 7875 7876 7883 7884 7906 7907 7911 7912 7913 7914
7915 7916 7917 7918 7919 7920 7921 7922 7923 7924 7925 7926 7932
7933 8005 8006 8010 8011 8028 8029 8030 8031
691# 703# 4989 4990
691# 705# 937 938 957 958 1829 1830 1853 1854 1877 1878 1902
1903 1927 1928 1952 1953 1977 1978 2002 2003 2026 2027 2050 2051
2067 2068 2091 2092 2116 2117 2133 2134 2150 2151 2174 2175 2191
2192 2208 2209 2233 2234 2250 2251 2274 2275 2298 2299 2323 2324
2347 2348 2372 2373 2396 2397 2413 2414 2430 2431 2454 2455 2478
2479 2502 2503 2526 2527 2546 2547 2559 2560 2709 2710 2730 2731
2750 2751 2768 2769 2816 2817 2838 2839 2887 2888 2911 2912 2914
2915 2960 2961 2963 2964 2993 2994 3010 3011 3014 3015 3044 3045
3058 3059 3062 3063 3092 3093 3106 3107 3110 3111 3156 3157 3159
3160 3196 3197 3221 3222 3224 3225 3260 3261 3286 3287 3289 3290
3327 3328 3348 3349 3351 3352 3383 3384 3401 3402 3404 3405 3440
3441 3481 3482 3510 3511 3513 3514 3554 3555 3586 3587 3589 3590
3629 3630 3658 3659 3661 3662 3701 3702 3730 3731 3733 3734 3773
3774 3802 3803 3805 3806 3845 3846 3874 3875 3877 3878 3919 3920
3950 3951 3953 3954 3996 3997 4028 4029 4031 4032 4071 4072 4100
4101 4103 4104 4143 4144 4172 4173 4175 4176 4215 4216 4244 4245
4247 4248 4287 4288 4316 4317 4319 4320 4359 4360 4388 4389 4391
4392 4431 4432 4460 4461 4463 4464 4503 4504 4532 4533 4535 4536
4575 4576 4604 4605 4607 4608 4656 4657 4685 4686 4688 4689 4766
4767 4882 4883 4922 4923 4968 4969 4971 4972 5020 5021 5055 5056
5062 5063 5065 5066 5113 5114 5143 5144 5146 5147 5190 5191 5232
5233 5281 5282 5336 5337 5385 5386 5432 5433 5478 5479 5537 5538
5590 5591 5646 5647 5687 5688 5753 5754 5799 5800 5802 5803 5860
5861 5871 5872 5924 5925 5935 5936 5988 5989 5999 6000 6052 6053

```
                    6063  6064  6116  6117  6127  6128  6180  6181  6191  6192  6244  6245  6255
                    6256  6308  6309  6319  6320  6372  6373  6384  6385  6438  6439  6449  6450
                    6503  6504  6515  6516  6569  6570  6580  6581  6634  6635  6645  6646  6699
                    6700  6710  6711  6764  6765  6775  6776  6830  6831  6841  6842  6895  6896
                    6906  6907  6961  6962  6972  6973  7026  7027  7037  7038  7091  7092  7102
                    7103  7156  7157  7167  7168  7221  7222  7232  7233  7286  7287  7297  7298
                    7351  7352  7362  7363  7416  7417  7427  7428  7481  7482  7492  7493  7547
                    7548  7558  7559  7612  7613  7623  7624  7677  7678  7688  7689  7742  7743
                    7753  7754  7806  7807  7817  7818  7871  7872  7882  7883  7933  7934  8011
                    8012
SVCTST= 000000        691#  702#  2789  2790  2825  2826  2861  2862  2925  2926  2974  2975  3025
                    3026  3073  3074  3120  3121  3171  3172  3235  3236  3300  3301  3362  3363
                    3417  3418  3451  3452  3524  3525  3600  3601  3672  3673  3744  3745  3816
                    3817  3888  3889  3964  3965  4042  4043  4114  4115  4186  4187  4258  4259
                    4330  4331  4402  4403  4474  4475  4546  4547  4618  4619  4700  4701  4775
                    4776  4893  4894  4983  4984  5077  5078  5157  5158  5201  5202  5244  5245
                    5293  5294  5347  5348  5396  5397  5443  5444  5490  5491  5553  5554  5602
                    5603  5658  5659  5699  5700  5768  5769  5821  5822  5885  5886  5949  5950
                    6013  6014  6077  6078  6141  6142  6205  6206  6269  6270  6333  6334  6399
                    6400  6464  6465  6530  6531  6595  6596  6660  6661  6725  6726  6791  6792
                    6856  6857  6922  6923  6987  6988  7052  7053  7117  7118  7182  7183  7247
                    7248  7312  7313  7377  7378  7442  7443  7508  7509  7573  7574  7638  7639
                    7703  7704  7767  7768  7832  7833
SWPAC1  041443      7964#
SWPAC2  041504      7970#
S$LSYM= 010000        691#  938#  958#  1830#  1854#  1878#  1903#  1928#  1953#  1978#  2003#  2027#  2051#
                    2068#  2092#  2117#  2134#  2151#  2175#  2192#  2209#  2234#  2251#  2275#  2299#  2324#
                    2348#  2373#  2397#  2414#  2431#  2455#  2479#  2503#  2527#  2547#  2560#  2710#  2731#
                    2751#  2769#  2817#  2839#  2867#  2893#  2915#  2930#  2964#  2977#  2996#  3015#  3028#
                    3047#  3063#  3076#  3095#  3111#  3125#  3160#  3177#  3200#  3225#  3241#  3264#  3290#
                    3305#  3330#  3352#  3367#  3387#  3405#  3441#  3458#  3485#  3514#  3531#  3558#  3590#
                    3606#  3633#  3662#  3678#  3705#  3734#  3750#  3777#  3806#  3822#  3849#  3878#  3894#
                    3923#  3954#  3970#  4000#  4032#  4048#  4075#  4104#  4120#  4147#  4176#  4192#  4219#
                    4248#  4264#  4291#  4320#  4336#  4363#  4392#  4408#  4435#  4464#  4480#  4507#  4536#
                    4552#  4579#  4608#  4624#  4661#  4689#  4767#  4883#  4899#  4927#  4972#  4992#  5024#
                    5063#  5066#  5083#  5118#  5147#  5191#  5233#  5282#  5337#  5386#  5433#  5479#  5538#
                    5591#  5647#  5688#  5754#  5776#  5803#  5831#  5872#  5895#  5936#  5959#  6000#  6023#
                    6064#  6087#  6128#  6151#  6192#  6215#  6256#  6279#  6320#  6343#  6385#  6409#  6450#
                    6474#  6516#  6540#  6581#  6605#  6646#  6670#  6711#  6735#  6776#  6801#  6842#  6866#
                    6907#  6932#  6973#  6997#  7038#  7062#  7103#  7127#  7168#  7192#  7233#  7257#  7298#
                    7322#  7363#  7387#  7428#  7452#  7493#  7518#  7559#  7583#  7624#  7648#  7689#  7713#
                    7754#  7777#  7818#  7842#  7883#  7934#  8012#
TDATA   030532      5772  5774  5804#
TEMP    002416      1086#
TESTAD  003224      1299*  1305#
TFM1    004277      1563#  1897  1922  1947  1972  1997  2228
TFM2    004330      1568#  1824  1848  1872  2021  2045  2086  2169  2269  2293  2342  2391  2449
                    2473  2497  2521
TFM27   004362      1574#  2111  2318  2367
TFM5    004345      1571#
TOUTP   027552      5579  5587#
TOUTT   027546      5556  5586#
TYPE    002602      1119#  2696*  2701*  2706*
T$ARGC= 000001        725#  726#  727#  728#  729#  730#  1809#  1815  1816#  1821  1822#  1828  1833#
                    1839  1840#  1845  1846#  1852  1857#  1863  1864#  1869  1870#  1876  1881#  1887
                    1888#  1893  1894#  1901  1906#  1912  1913#  1918  1919#  1926  1931#  1937  1938#
```

1943 1944# 1951 1956# 1962 1963# 1968 1969# 1976 1981# 1987 1988# 1993
1994# 2001 2006# 2012 2013# 2018 2019# 2025 2030# 2036 2037# 2042 2043#
2049 2054# 2060 2061# 2066 2071# 2077 2078# 2083 2084# 2090 2095# 2101
2102# 2107 2108# 2115 2120# 2126 2127# 2132 2137# 2143 2144# 2149 2154#
2160 2161# 2166 2167# 2173 2178# 2184 2185# 2190 2195# 2201 2202# 2207
2212# 2218 2219# 2224 2225# 2232 2237# 2243 2244# 2249 2254# 2260 2261#
2266 2267# 2273 2278# 2284 2285# 2290 2291# 2297 2302# 2308 2309# 2314
2315# 2322 2327# 2333 2334# 2339 2340# 2346 2351# 2357 2358# 2363 2364#
2371 2376# 2382 2383# 2388 2389# 2395 2400# 2406 2407# 2412 2417# 2423
2424# 2429 2434# 2440 2441# 2446 2447# 2453 2458# 2464 2465# 2470 2471#
2477 2482# 2488 2489# 2494 2495# 2501 2506# 2512 2513# 2518 2519# 2525
4869# 4873 4876# 4880

T$CODE= 003032
T$ERRN= 000027

7911# 7916# 7921#
691# 1480# 2806# 2833# 2876# 2900# 2939# 2951# 2986# 3003# 3037# 3054# 3085#
3102# 3135# 3147# 3186# 3210# 3250# 3275# 3314# 3337# 3376# 3394# 3431# 3471#
3499# 3547# 3575# 3619# 3647# 3691# 3719# 3763# 3791# 3835# 3863# 3909# 3939#
3986# 4017# 4061# 4089# 4133# 4161# 4205# 4233# 4277# 4305# 4349# 4377# 4421#
4449# 4493# 4521# 4565# 4593# 4645# 4673# 4738# 4750# 4762# 4857# 4912# 4941#
4956# 5010# 5044# 5102# 5132# 5178# 5184# 5220# 5226# 5275# 5321# 5329# 5373#
5420# 5466# 5532# 5582# 5626# 5640# 5682# 5727# 5792# 5849# 5913# 5977# 6041#
6105# 6169# 6233# 6297# 6361# 6427# 6492# 6558# 6623# 6688# 6753# 6819# 6884#
6950# 7015# 7080# 7145# 7210# 7275# 7340# 7405# 7470# 7536# 7601# 7666# 7731#
7795# 7860#

T$EXCP= 000000
T$FLAG= 000040

7911# 7915 7916# 7920 7921# 7926
2541# 2543 2797# 2813# 2880# 2904# 2943# 2955# 2990# 3007# 3041# 3089# 3139#
3151# 3190# 3214# 3254# 3279# 3318# 3341# 3380# 3398# 3436# 3475# 3503# 3579#
3623# 3651# 3695# 3723# 3767# 3795# 3839# 3867# 3913# 3943# 3990# 4021# 4065#
4093# 4137# 4165# 4209# 4237# 4281# 4309# 4353# 4381# 4425# 4453# 4497# 4525#
4569# 4597# 4649# 4677# 4861# 4916# 4945# 5014# 5048# 5106# 5136# 5377# 5380#
5424# 5427# 5470# 5473# 5731# 5747# 5796# 5853# 5863# 5917# 5927# 5981# 5991#
6045# 6055# 6109# 6119# 6173# 6183# 6237# 6247# 6301# 6311# 6365# 6375# 6431#
6441# 6496# 6506# 6562# 6572# 6627# 6637# 6692# 6702# 6757# 6767# 6823# 6833#
6888# 6898# 6954# 6964# 7019# 7029# 7084# 7094# 7149# 7159# 7214# 7224# 7279#
7289# 7344# 7354# 7409# 7419# 7474# 7484# 7540# 7550# 7605# 7615# 7670# 7680#
7735# 7745# 7799# 7809# 7864# 7874#

T$GMAN= 000000
T$HILI= 000007
T$LAST= 000001
T$LOLI= 000004
T$LSYM= 010000

691#
7911# 7914 7916# 7919 7921# 7925
691# 8029#
7911# 7913 7916# 7918 7921# 7924
691# 938 958 1830 1854 1878 1903 1928 1953 1978 2003 2027 2051
2068 2092 2117 2134 2151 2175 2192 2209 2234 2251 2275 2299 2324
2348 2373 2397 2414 2431 2455 2479 2503 2527 2547 2560 2710 2731
2751 2769 2817 2839 2915 2964 3015 3063 3111 3160 3225 3290 3352
3405 3441 3514 3590 3662 3734 3806 3878 3954 4032 4104 4176 4248
4320 4392 4464 4536 4608 4689 4767 4883 4972 5063 5066 5147 5191
5233 5282 5337 5386 5433 5479 5538 5591 5647 5688 5754 5803 5872
5936 6000 6064 6128 6192 6256 6320 6385 6450 6516 6581 6646 6711
6776 6842 6907 6973 7038 7103 7168 7233 7298 7363 7428 7493 7559
7624 7689 7754 7818 7883 7934 8012
8032#

T$LTNO= 000120
T$NEST= 177777

691# 697# 922# 937# 951# 957# 1808# 1829# 1832# 1853# 1856# 1877# 1880#
1902# 1905# 1927# 1930# 1952# 1955# 1977# 1980# 2002# 2005# 2026# 2029# 2050#
2053# 2067# 2070# 2091# 2094# 2116# 2119# 2133# 2136# 2150# 2153# 2174# 2177#
2191# 2194# 2208# 2211# 2233# 2236# 2250# 2253# 2274# 2277# 2298# 2301# 2323#
2326# 2347# 2350# 2372# 2375# 2396# 2399# 2413# 2416# 2430# 2433# 2454# 2457#
2478# 2481# 2502# 2505# 2526# 2537# 2546# 2550# 2555# 2557# 2559# 2572# 2709#

```
                2726#  2730#  2745#  2750#  2766#  2768#  2790#  2816#  2826#  2838#  2862#  2867#  2887#
                2893#  2911#  2914#  2926#  2930#  2960#  2963#  2975#  2977#  2993#  2996#  3010#  3014#
                3026#  3028#  3044#  3047#  3058#  3062#  3074#  3076#  3092#  3095#  3106#  3110#  3121#
                3125#  3156#  3159#  3172#  3177#  3196#  3200#  3221#  3224#  3236#  3241#  3260#  3264#
                3286#  3289#  3301#  3305#  3327#  3330#  3348#  3351#  3363#  3367#  3383#  3387#  3401#
                3404#  3418#  3440#  3452#  3458#  3481#  3485#  3510#  3513#  3525#  3531#  3554#  3558#
                3586#  3589#  3601#  3606#  3629#  3633#  3658#  3661#  3673#  3678#  3701#  3705#  3730#
                3733#  3745#  3750#  3773#  3777#  3802#  3805#  3817#  3822#  3845#  3849#  3874#  3877#
                3889#  3894#  3919#  3923#  3950#  3953#  3965#  3970#  3996#  4000#  4028#  4031#  4043#
                4048#  4071#  4075#  4100#  4103#  4115#  4120#  4143#  4147#  4172#  4175#  4187#  4192#
                4215#  4219#  4244#  4247#  4259#  4264#  4287#  4291#  4316#  4319#  4331#  4336#  4359#
                4363#  4388#  4391#  4403#  4408#  4431#  4435#  4460#  4463#  4475#  4480#  4503#  4507#
                4532#  4535#  4547#  4552#  4575#  4579#  4604#  4607#  4619#  4624#  4656#  4661#  4685#
                4688#  4701#  4766#  4776#  4882#  4894#  4899#  4922#  4927#  4968#  4971#  4984#  4990#
                4992#  5020#  5024#  5055#  5062#  5065#  5078#  5083#  5113#  5118#  5143#  5146#  5158#
                5190#  5202#  5232#  5245#  5281#  5294#  5336#  5348#  5385#  5397#  5432#  5444#  5478#
                5491#  5537#  5554#  5590#  5603#  5646#  5659#  5687#  5700#  5753#  5769#  5776#  5799#
                5802#  5822#  5831#  5860#  5871#  5886#  5895#  5924#  5935#  5950#  5959#  5988#  5999#
                6014#  6023#  6052#  6063#  6078#  6087#  6116#  6127#  6142#  6151#  6180#  6191#  6206#
                6215#  6244#  6255#  6270#  6279#  6308#  6319#  6334#  6343#  6372#  6384#  6400#  6409#
                6438#  6449#  6465#  6474#  6503#  6515#  6531#  6540#  6569#  6580#  6596#  6605#  6634#
                6645#  6661#  6670#  6699#  6710#  6726#  6735#  6764#  6775#  6792#  6801#  6830#  6841#
                6857#  6866#  6895#  6906#  6923#  6932#  6961#  6972#  6988#  6997#  7026#  7037#  7053#
                7062#  7091#  7102#  7118#  7127#  7156#  7167#  7183#  7192#  7221#  7232#  7248#  7257#
                7286#  7297#  7313#  7322#  7351#  7362#  7378#  7387#  7416#  7427#  7443#  7452#  7481#
                7492#  7509#  7518#  7547#  7558#  7574#  7583#  7612#  7623#  7639#  7648#  7677#  7688#
                7704#  7713#  7742#  7753#  7768#  7777#  7806#  7817#  7833#  7842#  7871#  7882#  7906#
                7932#  8005#  8010#  8020#
T$NS0 = 000000   697#  8020
T$NS1 = 000005   922#   937  951#   957  1808#  1829  1832#  1853  1856#  1877  1880#  1902  1905#
                1927  1930#  1952  1955#  1977  1980#  2002  2005#  2026  2029#  2050  2053#  2067
                2070#  2091  2094#  2116  2119#  2133  2136#  2150  2153#  2174  2177#  2191  2194#
                2208  2211#  2233  2236#  2250  2253#  2274  2277#  2298  2301#  2323  2326#  2347
                2350#  2372  2375#  2396  2399#  2413  2416#  2430  2433#  2454  2457#  2478  2481#
                2502  2505#  2526  2537#  2546  2550#  2555  2557#  2559  2572#  2709  2726#  2730
                2745#  2750  2766#  2768  2790#  2816  2826#  2838  2862#  2914  2926#  2963  2975#
                3014  3026#  3062  3074#  3110  3121#  3159  3172#  3224  3236#  3289  3301#  3351
                3363#  3404  3418#  3440  3452#  3513  3525#  3589  3601#  3661  3673#  3733  3745#
                3805  3817#  3877  3889#  3953  3965#  4031  4043#  4103  4115#  4175  4187#  4247
                4259#  4319  4331#  4391  4403#  4463  4475#  4535  4547#  4607  4619#  4688  4701#
                4766  4776#  4882  4894#  4971  4984#  5065  5078#  5146  5158#  5190  5202#  5232
                5245#  5281  5294#  5336  5348#  5385  5397#  5432  5444#  5478  5491#  5537  5554#
                5590  5603#  5646  5659#  5687  5700#  5753  5769#  5802  5822#  5871  5886#  5935
                5950#  5999  6014#  6063  6078#  6127  6142#  6191  6206#  6255  6270#  6319  6334#
                6384  6400#  6449  6465#  6515  6531#  6580  6596#  6645  6661#  6710  6726#  6775
                6792#  6841  6857#  6906  6923#  6972  6988#  7037  7053#  7102  7118#  7167  7183#
                7232  7248#  7297  7313#  7362  7378#  7427  7443#  7492  7509#  7558  7574#  7623
                7639#  7688  7704#  7753  7768#  7817  7833#  7882  7906#  7932  8005#  8010
T$NS2 = 000003   2867#  2887  2893#  2911  2930#  2960  2977#  2993  2996#  3010  3028#  3044  3047#
                3058  3076#  3092  3095#  3106  3125#  3156  3177#  3196  3200#  3221  3241#  3260
                3264#  3286  3305#  3327  3330#  3348  3367#  3383  3387#  3401  3458#  3481  3485#
                3510  3531#  3554  3558#  3586  3606#  3629  3633#  3658  3678#  3701  3705#  3730
                3750#  3773  3777#  3802  3822#  3845  3849#  3874  3894#  3919  3923#  3950  3970#
                3996  4000#  4028  4048#  4071  4075#  4100  4120#  4143  4147#  4172  4192#  4215
                4219#  4244  4264#  4287  4291#  4316  4336#  4359  4363#  4388  4408#  4431  4435#
                4460  4480#  4503  4507#  4532  4552#  4575  4579#  4604  4624#  4656  4661#  4685
```

T$NS3 = 000003
T$PTNU= 000000
T$SAVL= 177777
T$SEGL= 177777
T$SEK0= 010000

```
 4899#  4922   4927#  4968   4990#  5062   5083#  5113   5118#  5143   5776#  5799   5831#
 5860   5895#  5924   5959#  5988   6023#  6052   6087#  6116   6151#  6180   6215#  6244
 6279#  6308   6343#  6372   6409#  6438   6474#  6503   6540#  6569   6605#  6634   6670#
 6699   6735#  6764   6801#  6830   6866#  6895   6932#  6961   6997#  7026   7062#  7091
 7127#  7156   7192#  7221   7257#  7286   7322#  7351   7387#  7416   7452#  7481   7518#
 7547   7583#  7612   7648#  7677   7713#  7742   7777#  7806   7842#  7871
 4992#  5020   5024#  5055
  691#
  691#
  691#  2867#  2881   2887#  2889   2893#  2905   2911#  2913   2930#  2944   2956   2960#
 2962   2977#  2991   2993#  2995   2996#  3008   3010#  3012   3028#  3042   3044#  3046
 3047#  3058#  3060   3076#  3090   3092#  3094   3095#  3106#  3108   3125#  3140   3152
 3156#  3158   3177#  3191   3196#  3198   3200#  3215   3221#  3223   3241#  3255   3260#
 3262   3264#  3280   3286#  3288   3305#  3319   3327#  3329   3330#  3342   3348#  3350
 3367#  3381   3383#  3385   3387#  3399   3401#  3403   3458#  3476   3481#  3483   3485#
 3504   3510#  3512   3531#  3554#  3556   3558#  3580   3586#  3588   3606#  3624   3629#
 3631   3633#  3652   3658#  3660   3678#  3696   3701#  3703   3705#  3724   3730#  3732
 3750#  3768   3773#  3775   3777#  3796   3802#  3804   3822#  3840   3845#  3847   3849#
 3868   3874#  3876   3894#  3914   3919#  3921   3923#  3944   3950#  3952   3970#  3991
 3996#  3998   4000#  4022   4028#  4030   4048#  4066   4071#  4073   4075#  4094   4100#
 4102   4120#  4138   4143#  4145   4147#  4166   4172#  4174   4192#  4210   4215#  4217
 4219#  4238   4244#  4246   4264#  4282   4287#  4289   4291#  4310   4316#  4318   4336#
 4354   4359#  4361   4363#  4382   4388#  4390   4408#  4426   4431#  4433   4435#  4454
 4460#  4462   4480#  4498   4503#  4505   4507#  4526   4532#  4534   4552#  4570   4575#
 4577   4579#  4598   4604#  4606   4624#  4650   4656#  4658   4661#  4678   4685#  4687
 4899#  4917   4922#  4924   4927#  4946   4968#  4970   4992#  5015   5020#  5022   5024#
 5055#  5057   5083#  5107   5113#  5115   5118#  5137   5143#  5145   5776#  5797   5799#
 5801   5831#  5854   5860#  5862   5895#  5918   5924#  5926   5959#  5982   5988#  5990
 6023#  6046   6052#  6054   6087#  6110   6116#  6118   6151#  6174   6180#  6182   6215#
 6238   6244#  6246   6279#  6302   6308#  6310   6343#  6366   6372#  6374   6409#  6432
 6438#  6440   6474#  6497   6503#  6505   6540#  6563   6569#  6571   6605#  6628   6634#
 6636   6670#  6693   6699#  6701   6735#  6758   6764#  6766   6801#  6824   6830#  6832
 6866#  6889   6895#  6897   6932#  6955   6961#  6963   6997#  7020   7026#  7028   7062#
 7085   7091#  7093   7127#  7150   7156#  7158   7192#  7215   7221#  7223   7257#  7280
 7286#  7288   7322#  7345   7351#  7353   7387#  7410   7416#  7418   7452#  7475   7481#
 7483   7518#  7541   7547#  7549   7583#  7606   7612#  7614   7648#  7671   7677#  7679
 7713#  7736   7742#  7744   7777#  7800   7806#  7808   7842#  7865   7871#  7873
 2867#  2881   2887   2893#  2905   2911   2930#  2944   2956   2960   2977#  2991   2993
 2996#  3008   3010   3028#  3042   3044   3047#  3058   3076#  3090   3092   3095#  3106
 3125#  3140   3152   3156   3177#  3191   3196   3200#  3215   3221   3241#  3255   3260
 3264#  3280   3286   3305#  3319   3327   3330#  3342   3348   3367#  3381   3383   3387#
 3399   3401   3458#  3476   3481   3485#  3504   3510   3531#  3554   3558#  3580   3586
 3606#  3624   3629   3633#  3652   3658   3678#  3696   3701   3705#  3724   3730   3750#
 3768   3773   3777#  3796   3802   3822#  3840   3845   3849#  3868   3874   3894#  3914
 3919   3923#  3944   3950   3970#  3991   3996   4000#  4022   4028   4048#  4066   4071
 4075#  4094   4100   4120#  4138   4143   4147#  4166   4172   4192#  4210   4215   4219#
 4238   4244   4264#  4282   4287   4291#  4310   4316   4336#  4354   4359   4363#  4382
 4388   4408#  4426   4431   4435#  4454   4460   4480#  4498   4503   4507#  4526   4532
 4552#  4570   4575   4579#  4598   4604   4624#  4650   4656   4661#  4678   4685   4899#
 4917   4922   4927#  4946   4968   4992#  5015   5020   5024#  5055   5083#  5107   5113
 5118#  5137   5143   5776#  5797   5799   5831#  5854   5860   5895#  5918   5924   5959#
 5982   5988   6023#  6046   6052   6087#  6110   6116   6151#  6174   6180   6215#  6238
 6244   6279#  6302   6308   6343#  6366   6372   6409#  6432   6438   6474#  6497   6503
 6540#  6563   6569   6605#  6628   6634   6670#  6693   6699   6735#  6758   6764   6801#
 6824   6830   6866#  6889   6895   6932#  6955   6961   6997#  7020   7026   7062#  7085
 7091   7127#  7150   7156   7192#  7215   7221   7257#  7280   7286   7322#  7345   7351
```

```
                    7387#  7410   7416   7452#  7475   7481   7518#  7541   7547   7583#  7606   7612   7648#
                    7671   7677   7713#  7736   7742   7777#  7800   7806   7842#  7865   7871
T$SUBN= 000000       691#  2789#  2825#  2861#  2925#  2974#  3025#  3073#  3120#  3171#  3235#  3300#  3362#
                    3417#  3451#  3524#  3600#  3672#  3744#  3816#  3888#  3964#  4042#  4114#  4186#  4258#
                    4330#  4402#  4474#  4546#  4618#  4700#  4775#  4893#  4983#  4989#  5077#  5157#  5201#
                    5244#  5293#  5347#  5396#  5443#  5490#  5553#  5602#  5658#  5699#  5768#  5821#  5885#
                    5949#  6013#  6077#  6141#  6205#  6269#  6333#  6399#  6464#  6530#  6595#  6660#  6725#
                    6791#  6856#  6922#  6987#  7052#  7117#  7182#  7247#  7312#  7377#  7442#  7508#  7573#
                    7638#  7703#  7767#  7832#
T$TAGL= 177777       691#
T$TAGN= 010174       691#   922#   951#  1808#  1832#  1856#  1880#  1905#  1930#  1955#  1980#  2005#  2029#
                    2053#  2070#  2094#  2119#  2136#  2153#  2177#  2194#  2211#  2236#  2253#  2277#  2301#
                    2326#  2350#  2375#  2399#  2416#  2433#  2457#  2481#  2505#  2537#  2550#  2557#  2572#
                    2726#  2745#  2766#  2790#  2826#  2862#  2926#  2975#  3026#  3074#  3121#  3172#  3236#
                    3301#  3363#  3418#  3452#  3525#  3601#  3673#  3745#  3817#  3889#  3965#  4043#  4115#
                    4187#  4259#  4331#  4403#  4475#  4547#  4619#  4701#  4776#  4894#  4984#  4990#  5078#
                    5158#  5202#  5245#  5294#  5348#  5397#  5444#  5491#  5554#  5603#  5659#  5700#  5769#
                    5822#  5886#  5950#  6014#  6078#  6142#  6206#  6270#  6334#  6400#  6465#  6531#  6596#
                    6661#  6726#  6792#  6857#  6923#  6988#  7053#  7118#  7183#  7248#  7313#  7378#  7443#
                    7509#  7574#  7639#  7704#  7768#  7833#  7906#  8005#
T$TEMP= 000000       826#   827#   828#   829#   830#   831#   832#   833#   834#   835#   836#   837#   838#
                     839#   840#   841#   842#   843#   844#   845#   846#   847#   848#   849#   850#   851#
                     852#   853#   854#   855#   856#   857#   858#   859#   860#   861#   862#   863#   864#
                     865#   866#   867#   868#   869#   870#   871#   872#   873#   874#   875#   876#   877#
                     878#   879#   880#   881#   882#   883#   884#   885#   886#   887#   888#   889#   890#
                     891#   892#   893#   894#   895#   896#   897#   898#   899#   900#   901#   902#   903#
                    2026#  2050#  2067#  2091#  2116#  2133#  2150#  2174#  2191#  2208#  2233#  2250#  2274#
                    2298#  2323#  2347#  2372#  2396#  2413#  2430#  2454#  2478#  2502#  2526#  2541#  2542
                    2546#  2555#  2559#  2709#  2730#  2750#  2768#  2797#  2798   2813#  2814   2816#  2838#
                    2880#  2881#  2887#  2904#  2905#  2911#  2914#  2943#  2944#  2955#  2956#  2960#  2963#
                    2990#  2991#  2993#  3007#  3008#  3010#  3014#  3041#  3042#  3044#  3058#  3062#  3089#
                    3090#  3092#  3106#  3110#  3139#  3140#  3151#  3152#  3156#  3159#  3190#  3191#  3196#
                    3214#  3215#  3221#  3224#  3254#  3255#  3260#  3279#  3280#  3286#  3289#  3318#  3319#
                    3327#  3341#  3342#  3348#  3351#  3380#  3381#  3383#  3398#  3399#  3401#  3404#  3436#
                    3437   3440#  3475#  3476#  3481#  3503#  3504#  3510#  3513#  3554#  3579#  3580#  3586#
                    3589#  3623#  3624#  3629#  3651#  3652#  3658#  3661#  3695#  3696#  3701#  3723#  3724#
                    3730#  3733#  3767#  3768#  3773#  3795#  3796#  3802#  3805#  3839#  3840#  3845#  3867#
                    3868#  3874#  3877#  3913#  3914#  3919#  3943#  3944#  3950#  3953#  3990#  3991#  3996#
                    4021#  4022#  4028#  4031#  4065#  4066#  4071#  4093#  4094#  4100#  4103#  4137#  4138#
                    4143#  4165#  4166#  4172#  4175#  4209#  4210#  4215#  4237#  4238#  4244#  4247#  4281#
                    4282#  4287#  4309#  4310#  4316#  4319#  4353#  4354#  4359#  4381#  4382#  4388#  4391#
                    4425#  4426#  4431#  4453#  4454#  4460#  4463#  4497#  4498#  4503#  4525#  4526#  4532#
                    4535#  4569#  4570#  4575#  4597#  4598#  4604#  4607#  4649#  4650#  4656#  4677#  4678#
                    4685#  4688#  4766#  4861#  4862   4882#  4916#  4917#  4922#  4945#  4946#  4968#  4971#
                    5014#  5015#  5020#  5048#  5049   5055#  5062#  5065#  5106#  5107#  5113#  5136#  5137#
                    5143#  5146#  5190#  5232#  5281#  5336#  5377#  5378   5380#  5381   5385#  5424#  5425
                    5427#  5428   5432#  5470#  5471   5473#  5474   5478#  5537#  5590#  5646#  5687#  5731#
                    5732   5747#  5748   5753#  5796#  5797#  5799#  5802#  5853#  5854#  5860#  5863#  5864
                    5871#  5917#  5918#  5924#  5927#  5928   5935#  5981#  5982#  5988#  5991#  5992   5999#
                    6045#  6046#  6052#  6055#  6056   6063#  6109#  6110#  6116#  6119#  6120   6127#  6173#
                    6174#  6180#  6183#  6184   6191#  6237#  6238#  6244#  6247#  6248   6255#  6301#  6302#
                    6308#  6311#  6312   6319#  6365#  6366#  6372#  6375#  6376   6384#  6431#  6432#  6438#
                    6441#  6442   6449#  6496#  6497#  6503#  6506#  6507   6515#  6562#  6563#  6569#  6572#
                    6573   6580#  6627#  6628#  6634#  6637#  6638   6645#  6692#  6693#  6699#  6702#  6703
                    6710#  6757#  6758#  6764#  6767#  6768   6775#  6823#  6824#  6830#  6833#  6834   6841#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0202

	6888#	6889#	6895#	6898#	6899	6906#	6954#	6955#	6961#	6964#	6965	6972#	7019#
	7020#	7026#	7029#	7030	7037#	7084#	7085#	7091#	7094#	7095	7102#	7149#	7150#
	7156#	7159#	7160	7167#	7214#	7215#	7221#	7224#	7225	7232#	7279#	7280#	7286#
	7289#	7290	7297#	7344#	7345#	7351#	7354#	7355	7362#	7409#	7410#	7416#	7419#
	7420	7427#	7474#	7475#	7481#	7484#	7485	7492#	7540#	7541#	7547#	7550#	7551
	7558#	7605#	7606#	7612#	7615#	7616	7623#	7670#	7671#	7677#	7680#	7681	7688#
	7735#	7736#	7742#	7745#	7746	7753#	7799#	7800#	7806#	7809#	7810	7817#	7864#
	7865#	7871#	7874#	7875	7882#	7911#	7916#	7921#	7932#	8010#	8020#		
T$TEST= 000120	691#	2782	2786	2789#	2819	2822	2825#	2841	2858	2861#	2917	2922	2925#
	2966	2971	2974#	3017	3022	3025#	3065	3070	3073#	3113	3117	3120#	3162
	3167	3171#	3227	3232	3235#	3292	3297	3300#	3354	3359	3362#	3407	3414
	3417#	3443	3448	3451#	3516	3521	3524#	3592	3597	3600#	3664	3669	3672#
	3736	3741	3744#	3808	3813	3816#	3880	3885	3888#	3956	3961	3964#	4034
	4039	4042#	4106	4111	4114#	4178	4183	4186#	4250	4255	4258#	4322	4327
	4330#	4394	4399	4402#	4466	4471	4474#	4538	4543	4546#	4610	4615	4618#
	4691	4697	4700#	4769	4772	4775#	4885	4890	4893#	4975	4980	4983#	4989
	5069	5074	5077#	5150	5154	5157#	5194	5198	5201#	5236	5241	5244#	5285
	5290	5293#	5340	5344	5347#	5389	5393	5396#	5436	5440	5443#	5482	5487
	5490#	5541	5550	5553#	5594	5599	5602#	5650	5655	5658#	5691	5696	5699#
	5761	5765	5768#	5811	5818	5821#	5875	5882	5885#	5939	5946	5949#	6003
	6010	6013#	6067	6074	6077#	6131	6138	6141#	6195	6202	6205#	6259	6266
	6269#	6323	6330	6333#	6389	6396	6399#	6454	6461	6464#	6520	6527	6530#
	6585	6592	6595#	6650	6657	6660#	6715	6722	6725#	6780	6788	6791#	6846
	6853	6856#	6911	6918	6922#	6977	6984	6987#	7042	7049	7052#	7107	7114
	7117#	7172	7180	7182#	7237	7244	7247#	7302	7309	7312#	7367	7374	7377#
	7432	7439	7442#	7497	7504	7508#	7563	7570	7573#	7628	7635	7638#	7693
	7700	7703#	7757	7764	7767#	7822	7829	7832#	8032				
T$TSTM= 177777	691#	1479	1814	1820	1827	1830	1838	1844	1851	1854	1862	1868	1875
	1878	1886	1892	1900	1903	1911	1917	1925	1928	1936	1942	1950	1953
	1961	1967	1975	1978	1986	1992	2000	2003	2011	2017	2024	2027	2035
	2041	2048	2051	2059	2065	2068	2076	2082	2089	2092	2100	2106	2114
	2117	2125	2131	2134	2142	2148	2151	2159	2165	2172	2175	2183	2189
	2192	2200	2206	2209	2217	2223	2231	2234	2242	2248	2251	2259	2265
	2272	2275	2283	2289	2296	2299	2307	2313	2321	2324	2332	2338	2345
	2348	2356	2362	2370	2373	2381	2387	2394	2397	2405	2411	2414	2422
	2428	2431	2439	2445	2452	2455	2463	2469	2476	2479	2487	2493	2500
	2503	2511	2517	2524	2527	2547	2560	2590	2599	2605	2612	2627	2710
	2731	2748	2751	2769	2797	2805	2813	2817	2832	2839	2867	2875	2880
	2888	2893	2899	2904	2912	2915	2930	2938	2943	2950	2955	2961	2964
	2977	2985	2990	2994	2996	3002	3007	3011	3015	3028	3036	3041	3045
	3047	3053	3059	3063	3076	3084	3089	3093	3095	3101	3107	3111	3125
	3134	3139	3146	3151	3157	3160	3177	3185	3190	3197	3200	3209	3214
	3222	3225	3241	3249	3254	3261	3264	3274	3279	3287	3290	3305	3313
	3318	3328	3330	3336	3341	3349	3352	3367	3375	3380	3384	3387	3393
	3398	3402	3405	3430	3436	3441	3458	3470	3475	3482	3485	3498	3503
	3511	3514	3531	3546	3555	3558	3574	3579	3587	3590	3606	3618	3623
	3630	3633	3646	3651	3659	3662	3678	3690	3695	3702	3705	3718	3723
	3731	3734	3750	3762	3767	3774	3777	3790	3795	3803	3806	3822	3834
	3839	3846	3849	3862	3867	3875	3878	3894	3908	3913	3920	3923	3938
	3943	3951	3954	3970	3985	3990	3997	4000	4016	4021	4029	4032	4048
	4060	4065	4072	4075	4088	4093	4101	4104	4120	4132	4137	4144	4147
	4160	4165	4173	4176	4192	4204	4209	4216	4219	4232	4237	4245	4248
	4264	4276	4281	4288	4291	4304	4309	4317	4320	4336	4348	4353	4360
	4363	4376	4381	4389	4392	4408	4420	4425	4432	4435	4448	4453	4461
	4464	4480	4492	4497	4504	4507	4520	4525	4533	4536	4552	4564	4569
	4576	4579	4592	4597	4605	4608	4624	4644	4649	4657	4661	4672	4677

```
                    4686  4689  4737  4749  4761  4767  4856  4861  4872  4879  4883  4899  4911
                    4916  4923  4927  4940  4945  4955  4960  4969  4972  4990  4992  5009  5014
                    5021  5024  5043  5048  5056  5063  5066  5083  5101  5106  5114  5118  5131
                    5136  5144  5147  5168  5174  5177  5183  5191  5210  5216  5219  5225  5233
                    5250  5267  5274  5282  5299  5317  5320  5328  5337  5372  5377  5380  5386
                    5419  5424  5427  5433  5465  5470  5473  5479  5531  5538  5581  5591  5625
                    5639  5647  5681  5688  5726  5731  5747  5754  5776  5791  5796  5800  5803
                    5831  5848  5853  5861  5863  5872  5895  5912  5917  5925  5927  5936  5959
                    5976  5981  5989  5991  6000  6023  6040  6045  6053  6055  6064  6087  6104
                    6109  6117  6119  6128  6151  6168  6173  6181  6183  6192  6215  6232  6237
                    6245  6247  6256  6279  6296  6301  6309  6311  6320  6343  6360  6365  6373
                    6375  6385  6409  6426  5431  6439  6441  6450  6474  6491  6496  6504  6506
                    6516  6540  6557  6562  6570  6572  6581  6605  6622  6627  6635  6637  6646
                    6670  6687  6692  6700  6702  6711  6735  6752  6757  6765  6767  6776  6801
                    6818  6823  6831  6833  6842  6866  6883  6888  6896  6898  6907  6932  6949
                    6954  6962  6964  6973  6997  7014  7019  7027  7029  7038  7062  7079  7084
                    7092  7094  7103  7127  7144  7149  7157  7159  7168  7192  7209  7214  7222
                    7224  7233  7257  7274  7279  7287  7289  7298  7322  7339  7344  7352  7354
                    7363  7387  7404  7409  7417  7419  7428  7452  7469  7474  7482  7484  7493
                    7518  7535  7540  7548  7550  7559  7583  7600  7605  7613  7615  7624  7648
                    7665  7670  7678  7680  7689  7713  7730  7735  7743  7745  7754  7777  7794
                    7799  7807  7809  7818  7842  7859  7864  7872  7874  7883
T$TSTS= 000001        691#  2790#  2826#  2862#  2926#  2975#  3026#  3074#  3121#  3172#  3236#  3301#  3363#
                    3418#  3452#  3525#  3601#  3673#  3745#  3817#  3889#  3965#  4043#  4115#  4187#  4259#
                    4331#  4403#  4475#  4547#  4619#  4701#  4776#  4894#  4984#  5078#  5158#  5202#  5245#
                    5294#  5348#  5397#  5444#  5491#  5554#  5603#  5659#  5700#  5769#  5822#  5886#  5950#
                    6014#  6078#  6142#  6206#  6270#  6334#  6400#  6465#  6531#  6596#  6661#  6726#  6792#
                    6857#  6923#  6988#  7053#  7118#  7183#  7248#  7313#  7378#  7443#  7509#  7574#  7639#
                    7704#  7768#  7833#
T$$AU  = 010050     2766#  2768
T$$AUT= 010044      2557#  2559
T$$CLE= 010046      2726#  2730
T$$DU  = 010047     2745#  2750
T$$HAR= 010172      7906#  7933
T$$HW  = 010000       922#   937
T$$INI= 010045      2572#  2709
T$$MSG= 010041      1808#  1829  1832#  1853  1856#  1877  1880#  1902  1905#  1927  1930#  1952  1955#
                    1977   1980#  2002  2005#  2026  2029#  2050  2053#  2067  2070#  2091  2094#  2116
                    2119#  2133  2136#  2150  2153#  2174  2177#  2191  2194#  2208  2211#  2233  2236#
                    2250   2253#  2274  2277#  2298  2301#  2323  2326#  2347  2350#  2372  2375#  2396
                    2399#  2413  2416#  2430  2433#  2454  2457#  2478  2481#  2502  2505#  2526
T$$PRO= 010043      2550#
T$$RPT= 010042      2537#  2541  2546
T$$SEG= 010000      2867#  2880  2887#  2893#  2904  2911#  2930#  2943  2955  2960#  2977#  2990  2993#
                    2996#  3007  3010#  3028#  3041  3044#  3047#  3058#  3076#  3089  3092#  3095#  3106#
                    3125#  3139  3151  3156#  3177#  3190  3196#  3200#  3214  3221#  3241#  3254  3260#
                    3264#  3279  3286#  3305#  3318  3327#  3330#  3341  3348#  3367#  3380  3383#  3387#
                    3398   3401#  3458#  3475  3481#  3485#  3503  3510#  3531#  3554#  3558#  3579  3586#
                    3606#  3623  3629#  3633#  3651  3658#  3678#  3695  3701#  3705#  3723  3730#  3750#
                    3767   3773#  3777#  3795  3802#  3822#  3839  3845#  3849#  3867  3874#  3894#  3913
                    3919#  3923#  3943  3950#  3970#  3990  3996#  4000#  4021  4028#  4048#  4065  4071#
                    4075#  4093  4100#  4120#  4137  4143#  4147#  4165  4172#  4192#  4209  4215#  4219#
                    4237   4244#  4264#  4281  4287#  4291#  4309  4316#  4336#  4353  4359#  4363#  4381
                    4388#  4408#  4425  4431#  4435#  4453  4460#  4480#  4497  4503#  4507#  4525  4532#
                    4552#  4569  4575#  4579#  4597  4604#  4624#  4649  4656#  4661#  4677  4685#  4899#
                    4916   4922#  4927#  4945  4968#  4992#  5014  5020#  5024#  5055#  5083#  5106  5113#
```

```
                    5118#  5136   5143#  5776#  5796   5799#  5831#  5853   5860#  5895#  5917   5924#  5959#
                    5981   5988#  6023#  6045   6052#  6087#  6109   6116#  6151#  6173   6180#  6215#  6237
                    6244#  6279#  6301   6308#  6343#  6365   6372#  6409#  6431   6438#  6474#  6496   6503#
                    6540#  6562   6569#  6605#  6627   6634#  6670#  6692   6699#  6735#  6757   6764#  6801#
                    6823   6830#  6866#  6888   6895#  6932#  6954   6961#  6997#  7019   7026#  7062#  7084
                    7091#  7127#  7149   7156#  7192#  7214   7221#  7257#  7279   7286#  7322#  7344   7351#
                    7387#  7409   7416#  7452#  7474   7481#  7518#  7540   7547#  7583#  7605   7612#  7648#
                    7670   7677#  7713#  7735   7742#  7777#  7799   7806#  7842#  7864   7871#
T$$SOF= 010173      8005#  8011
T$$SUB= 010113      4990#  5062
T$$SW = 010001       951#   957
T$$TES= 010171      2790#  2797   2813   2816   2826#  2838   2862#  2914   2926#  2963   2975#  3014   3026#
                    3062   3074#  3110   3121#  3159   3172#  3224   3236#  3289   3301#  3351   3363#  3404
                    3418#  3436   3440   3452#  3513   3525#  3589   3601#  3661   3673#  3733   3745#  3805
                    3817#  3877   3889#  3953   3965#  4031   4043#  4103   4115#  4175   4187#  4247   4259#
                    4319   4331#  4391   4403#  4463   4475#  4535   4547#  4607   4619#  4688   4701#  4766
                    4776#  4861   4882   4894#  4971   4984#  5048   5065   5078#  5146   5158#  5190   5202#
                    5232   5245#  5281   5294#  5336   5348#  5377   5380   5385   5397#  5424   5427   5432
                    5444#  5470   5473   5478   5491#  5537   5554#  5590   5603#  5646   5659#  5687   5700#
                    5731   5747   5753   5769#  5802   5822#  5863   5871   5886#  5927   5935   5950#  5991
                    5999   6014#  6055   6063   6078#  6119   6127   6142#  6183   6191   6206#  6247   6255
                    6270#  6311   6319   6334#  6375   6384   6400#  6441   6449   6465#  6506   6515   6531#
                    6572   6580   6596#  6637   6645   6661#  6702   6710   6726#  6767   6775   6792#  6833
                    6841   6857#  6898   6906   6923#  6964   6972   6988#  7029   7037   7053#  7094   7102
                    7118#  7159   7167   7183#  7224   7232   7248#  7289   7297   7313#  7354   7362   7378#
                    7419   7427   7443#  7484   7492   7509#  7550   7558   7574#  7615   7623   7639#  7680
                    7688   7704#  7745   7753   7768#  7809   7817   7833#  7874   7882
T1          013176 G     826    2789#
T10         014770 G     835    3235#
T11         015174 G     836    3300#
T12         015376 G     837    3362#
T13         015562 G     838    3417#
T14         015674 G     839    3451#
T15         016134 G     840    3524#
T16         016430 G     841    3600#
T17         016670 G     842    3672#
T18         017130 G     843    3744#
T19         017370 G     844    3816#
T2          013320 G     827    2825#
T20         017630 G     845    3888#
T21         020120 G     846    3964#
T22         020422 G     847    4042#
T23         020662 G     848    4114#
T24         021122 G     849    4186#
T25         021362 G     850    4258#
T26         021622 G     851    4330#
T27         022062 G     852    4402#
T28         022322 G     853    4474#
T29         022562 G     854    4546#
T3          013366 G     828    2861#
T30         023022 G     855    4618#
T31         023330 G     856    4700#
T32         023726 G     857    4775#
T33         024412 G     858    4893#
T34         024722 G     859    4983#
T34.1       024740       4989#
```

T35	025262 G	860	5077#				
T36	025570 G	861	5157#				
T37	025736 G	862	5201#				
T38	026100 G	863	5244#				
T39	026252 G	864	5293#				
T4	013576 G	829	2925#				
T40	026456 G	865	5347#				
T41	026632 G	866	5396#				
T42	027006 G	867	5443#				
T43	027156 G	868	5490#				
T44	027350 G	869	5553#				
T45	027570 G	870	5602#				
T46	030016 G	871	5658#				
T47	030154 G	872	5699#				
T48	030376 G	873	5768#				
T49	030542 G	874	5821#				
T5	013746 G	830	2974#				
T50	030752 G	875	5885#				
T51	031162 G	876	5949#				
T52	031372 G	877	6013#				
T53	031602 G	878	6077#				
T54	032012 G	879	6141#				
T55	032222 G	880	6205#				
T56	032432 G	881	6269#				
T57	032642 G	882	6333#				
T58	033054 G	883	6399#				
T59	033264 G	884	6464#				
T6	014112 G	831	3025#				
T60	033474 G	885	6530#				
T61	033704 G	886	6595#				
T62	034114 G	887	6660#				
T63	034324 G	888	6725#				
T64	034534 G	889	6791#				
T65	034744 G	890	6856#				
T66	035154 G	891	6922#				
T67	035364 G	892	6987#				
T68	035574 G	893	7052#				
T69	036004 G	894	7117#				
T7	014252 G	832	3073#				
T70	036214 G	895	7182#				
T71	036424 G	896	7247#				
T72	036634 G	897	7312#				
T73	037044 G	898	7377#				
T74	037254 G	899	7442#				
T75	037464 G	900	7508#				
T76	037674 G	901	7573#				
T77	040104 G	902	7638#				
T78	040314 G	903	7703#				
T79	040524 G	904	7767#				
T8	014412 G	833	3120#				
T80	040734 G	905	7832#				
T9	014564 G	834	3171#				
UAM =	000200 G	1043#					
VECTOR	041306	7917	7948#				
WMP	041200	7935#					
WTYPE	002600	1118#	2632*	2697	2699	2704	

```
X$ALWA= 000000   691#
X$FALS= 000040   691#
X$OFFS= 000400   691#
X$TRUE= 000020   691#
ZERO    002552  1106#
$BDADR  002610  1122#  1479*  2805*  2832*  2875*  2899*  2938*  2950*  2985*  3002*  3036*  3053*  3084*
                3101*  3134*  3146*  3185*  3209*  3249*  3274*  3313*  3336*  3375*  3393*  3430*  3470*
                3498*  3546*  3574*  3618*  3646*  3690*  3718*  3762*  3790*  3834*  3862*  3908*  3938*
                3985*  4016*  4060*  4088*  4132*  4160*  4204*  4232*  4276*  4304*  4348*  4376*  4420*
                4448*  4492*  4520*  4564*  4592*  4644*  4672*  4737*  4749*  4761*  4856*  4911*  4940*
                4955*  5009*  5043*  5101*  5131*  5177*  5183*  5219*  5225*  5274*  5320*  5328*  5372*
                5419*  5465*  5531*  5581*  5625*  5639*  5681*  5726*  5791*  5848*  5912*  5976*  6040*
                6104*  6168*  6232*  6296*  6360*  6426*  6491*  6557*  6622*  6687*  6752*  6818*  6883*
                6949*  7014*  7079*  7144*  7209*  7274*  7339*  7404*  7469*  7535*  7600*  7665*  7730*
                7794*  7859*
$BDDAT  002614  1124#  1479*  1822  1846  1870  1895  1920  1945  1970  1995  2019  2043  2084
                2108  2167  2226  2267  2291  2315  2340  2364  2389  2447  2471  2495  2519
                2805*  2832*  2875*  2899*  2938*  2950*  2985*  3002*  3036*  3053*  3084*  3101*  3134*
                3146*  3185*  3209*  3249*  3274*  3313*  3336*  3375*  3393*  3430*  3470*  3498*  3546*
                3574*  3618*  3646*  3690*  3718*  3762*  3790*  3834*  3862*  3908*  3938*  3985*  4016*
                4060*  4088*  4132*  4160*  4204*  4232*  4276*  4304*  4348*  4376*  4420*  4448*  4492*
                4520*  4564*  4592*  4644*  4672*  4737*  4749*  4761*  4856*  4911*  4940*  4955*  5009*
                5043*  5101*  5131*  5177*  5183*  5219*  5225*  5274*  5320*  5328*  5372*  5419*  5465*
                5531*  5581*  5625*  5639*  5681*  5726*  5791*  5848*  5912*  5976*  6040*  6104*  6168*
                6232*  6296*  6360*  6426*  6491*  6557*  6622*  6687*  6752*  6818*  6883*  6949*  7014*
                7079*  7144*  7209*  7274*  7339*  7404*  7469*  7535*  7600*  7665*  7730*  7794*  7859*
$GDADR  002606  1121#  1479*  1894  1919  1944  1969  1994  2225  2805*  2832*  2875*  2899*  2938*
                2950*  2985*  3002*  3036*  3053*  3084*  3101*  3134*  3146*  3185*  3209*  3249*  3274*
                3313*  3336*  3375*  3393*  3430*  3470*  3498*  3546*  3574*  3618*  3646*  3690*  3718*
                3762*  3790*  3834*  3862*  3908*  3938*  3985*  4016*  4060*  4088*  4132*  4160*  4204*
                4232*  4276*  4304*  4348*  4376*  4420*  4448*  4492*  4520*  4564*  4592*  4644*  4672*
                4737*  4749*  4761*  4856*  4911*  4940*  4955*  5009*  5043*  5101*  5131*  5177*  5183*
                5219*  5225*  5274*  5320*  5328*  5372*  5419*  5465*  5531*  5581*  5625*  5639*  5681*
                5726*  5791*  5848*  5912*  5976*  6040*  6104*  6168*  6232*  6296*  6360*  6426*  6491*
                6557*  6622*  6687*  6752*  6818*  6883*  6949*  7014*  7079*  7144*  7209*  7274*  7339*
                7404*  7469*  7535*  7600*  7665*  7730*  7794*  7859*
$GDDAT  002612  1123#  1474*  1476  1823  1847  1871  1896  1921  1946  1971  1996  2020  2044
                2085  2109  2168  2227  2268  2292  2316  2341  2365  2390  2448  2472  2496
                2520  2828*  2869*  2872  2894*  2896  2932*  2933  2935  2946*  2979*  2980  2982
                2998*  3030*  3031  3033  3049*  3078*  3079  3081  3097*  3128*  3129  3131  3142*
                3179*  3180  3182  3203*  3204  3206  3243*  3244  3246  3268*  3269  3271  3307*
                3310  3331*  3333  3369*  3370  3372  3385*  3388  3390  3425*  3427  3465*  3467
                3493*  3495  3539*  3541*  3543  3567*  3569*  3571  3613*  3615  3641*  3643  3685*
                3687  3713*  3715  3757*  3759  3785*  3787  3829*  3831  3857*  3859  3902*  3903*
                3905  3932*  3933*  3935  3978*  3979*  3982  4009*  4010*  4011*  4013  4055*  4057
                4083*  4085  4127*  4129  4155*  4157  4199*  4201  4227*  4229  4271*  4273  4299*
                4301  4343*  4345  4371*  4373  4415*  4417  4443*  4445  4487*  4489  4515*  4517
                4559*  4561  4587*  4589  4639*  4641  4667*  4669  4730*  4734  4742*  4746  4754*
                4758  4906*  4908  4935*  4937  5004*  5006  5038*  5040  5096*  5098  5126*  5128
                5257*  5367*  5369  5410*  5416  5460*  5462  5508*  5528  5576*  5577*  5578  5619*
                5622  5634*  5675*  5678  5721*  5723  5786*  5788  5843*  5845  5907*  5909  5971*
                5973  6035*  6037  6099*  6101  6163*  6165  6227*  6229  6291*  6293  6355*  6357
                6421*  6423  6486*  6488  6552*  6554  6617*  6619  6682*  6684  6747*  6749  6813*
                6815  6878*  6880  6944*  6946  7009*  7011  7074*  7076  7139*  7141  7204*  7206
                7269*  7271  7334*  7336  7399*  7401  7464*  7466  7530*  7532  7595*  7597  7660*
                7662  7725*  7727  7789*  7791  7854*  7856
```