

KMC11-B

KMC11-B STATIC PART 1 AH-5877A-MC  
CZK MBAO FICHE 1 OF 2

APR 1982  
COPYRIGHT © 1982  
MADE IN USA



The main body of the document consists of a dense grid of approximately 15 columns and 25 rows of small, repetitive data blocks. Each block contains technical information, likely related to the static part of the KMC11-B system. The text within these blocks is too small to be legible but appears to follow a consistent format, possibly including component identifiers, values, and test procedures. The overall layout is highly structured and repetitive, typical of a technical manual or data sheet for a complex electronic system.



KMC11-B

KMC11-B STATIC PART 1  
CZKMBAO

AH-S877A-MC  
FICHE 2 OF 2

APR 1982  
COPYRIGHT © 1982  
MADE IN USA



.NLIST SEQ,BIN,LOC  
.REM 8

IDENTIFICATION

PRODUCT CODE: AC-S875A-MC  
PRODUCT NAME: CZKBAO KMC11-B STATIC PART1  
PROGRAM DATE: SEPTEMBER 1981  
MAINTAINER: CSS/NSG DIAGNOSTICS

COPYRIGHT (C) 1982 BY  
DIGITAL EQUIPMENT CORPORATION,  
MAYNARD, MASSACHUSETTS.  
ALL RIGHTS RESERVED.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY  
BE USED AND COPIED ONLY IN ACCORDANCE WITH THE  
TERMS OF SUCH LICENSE AND WITH THE INCLUSION  
OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE  
MADE AVAILABLE TO ANY OTHER PERSON. NO TITLE TO  
AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT  
NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL  
EQUIPMENT CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF  
ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

## 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 : (0) 174100 ? <CR>

-----  
VECTOR ADDRESS : (0) 300 ? <CR>

-----  
PRIORITY LEVEL : (0) 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-LIS )

<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 EXPLICIT 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:

CZKMBAD 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 (174.00 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



```
681          .TITLE CZKMBAD KMC11-B STATIC PART1
682          002000          .=2000
683
684
685
686
687
688
689          .MCALL  SVC
690 002000          SVC          : INITIALIZE SUPERVISOR MACROS
691
692
693
694
695
696 002000          BGNMOD  CZKMB
697
698
699          000000          $LSTIN= 0
700          000000          $LSTTAG= 0
701          000000          SVCINS= 0          : LIST INSTRUCTIONS, SHIFTED RIGHT
702          000000          SVCTST= 0          : LIST TEST TAGS, SHIFTED RIGHT
703          000000          SVCSUB= 0          : LIST SUBTEST TAGS, SHIFTED RIGHT
704          000000          SVCGBL= 0          : LIST GLOBAL TAGS, SHIFTED RIGHT
705          000000          SVCTAG= 0          : LIST OTHER TAGS, SHIFTED RIGHT
706
707          :          CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
708          :          TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS.  CHANGE THE
709          :          SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS.  YOU MAY
710          :          CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
711
712
713          .ENABL  AMA
```

714  
 715  
 716  
 717  
 718  
 719  
 720 002000  
 721  
 722  
 723 002000  
 724 002000  
 725 002000 103  
 726 002001 132  
 727 002002 113  
 728 002003 115  
 729 002004 102  
 730 002005 000  
 731 002006 000  
 732 002007 000  
 733 002010  
 734 002010 101  
 735 002011  
 736 002011 060  
 737 002012  
 738 002012 000000  
 739 002014  
 740 002014 000360  
 741 002016  
 742 002016 041146  
 743 002020  
 744 002020 000000  
 745 002022  
 746 002022 002366  
 747 002024  
 748 002024 002412  
 749 002026  
 750 002026 041740  
 751 002030  
 752 002030 000000  
 753 002032  
 754 002032 000000  
 755 002034  
 756 002034 000000  
 757 002036  
 758 002036 000000  
 759 002040  
 760 002040 002124  
 761 002042  
 762 002042 000000  
 763 002044  
 764 002044 000000  
 765 002046  
 766 002046 000000  
 767 002050  
 768 002050 003  
 769 002051 003

.SBTTL PROGRAM HEADER

+++  
 : THE PROGRAM HEADER IS THE INTERFACE BETWEEN  
 : THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.  
 :--

POINTER BGNAU,BGNDU,BGNSW,BGNRPT

HEADER CZKMB,A,0,240.,0  
 LSNAME:: :DIAGNOSTIC NAME  
 .ASCII /C/  
 .ASCII /Z/  
 .ASCII /K/  
 .ASCII /M/  
 .ASCII /B/  
 .BYTE 0  
 .BYTE 0  
 .BYTE 0  
 LSREV:: :REVISION LEVEL  
 .ASCII /A/  
 LSDEPO:: :0  
 .ASCII /0/  
 LSUNIT:: :NUMBER OF UNITS  
 .WORD 0  
 LSTIML:: :LONGEST TEST TIME  
 .WORD 240.  
 LSHPCP:: :PTR. TO H.W. QUES.  
 .WORD LSHARD  
 LSSPCP:: :PTR. TO S.W. QUES.  
 .WORD 0  
 LSHPTP:: :PTR. TO DEF. H.W. PTABLE  
 .WORD LSHW  
 LSSPTP:: :PTR. TO S.W. PTABLE  
 .WORD LSSW  
 LSLADP:: :DIAG. END ADDRESS  
 .WORD L\$LAST  
 L\$STA:: :RESERVED FOR APT STATS  
 .WORD 0  
 L\$CO::  
 .WORD 0  
 L\$DTP:: :DIAGNOSTIC TYPE  
 .WORD 0  
 L\$APT:: :APT EXPANSION  
 .WORD 0  
 L\$DTP:: :PTR. TO DISPATCH TABLE  
 .WORD L\$DISPATCH  
 L\$PRIO:: :DIAGNOSTIC RUN PRIORITY  
 .WORD 0  
 L\$ENV:: :FLAGS DESCRIBE HOW IT WAS SETUP  
 .WORD 0  
 L\$EXP1:: :EXPANSION WORD  
 .WORD 0  
 L\$MREV:: :SVC REV AND EDIT #  
 .BYTE C\$REVISION  
 .BYTE C\$EDIT

770	002052		LSEF::		;DIAG. EVENT FLAGS
771	002052	000000			
772	002054	000000			
773	002056		LSSPC::		
774	002056	000000			
775	002060		L\$DEVP::		; POINTER TO DEVICE TYPE LIST
776	002060	003116			
777	002062		L\$REPP::		;PTR. TO REPORT CODE
778	002062	012454			
779	002064		L\$EXP4::		
780	002064	000000			
781	002066		L\$EXP5::		
782	002066	000000			
783	002070		L\$AUT::		;PTR. TO ADD UNIT CODE
784	002070	013174			
785	002072		L\$DUT::		;PTR. TO DROP UNIT CODE
786	002072	013170			
787	002074		L\$LUN::		;LUN FOR EXERCISERS TO FILL
788	002074	000000			
789	002076		L\$DESP::		;POINTER TO DIAG. DESCRIPTION
790	002076	003124			
791	002100		L\$LOAD::		;GENERATE SPECIAL AUTOLOAD EMT
792	002100	104035			
793	002102				
794	002102	000000			
795	002104		L\$ICP::		;PTR. TO INIT CODE
796	002104	012472			
797	002106		L\$CCP::		;PTR. TO CLEAN-UP CODE
798	002106	013166			
799	002110		L\$ACP::		;PTR. TO AUTO CODE
800	002110	012470			
801	002112		L\$PRT::		;PTR. TO PROTECT TABLE
802	002112	012462			
803	002114		L\$TEST::		;TEST NUMBER
804	002114	000000			
805	002116		L\$DLY::		;DELAY COUNT
806	002116	000000			
807	002120		L\$HIME::		;PTR. TO HIGH MEM
808	002120	000000			
809					
810					
811					
812					
813					
814					
815					



.SBTTL DISPATCH TABLE

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

816  
817  
818  
819  
820  
821  
822  
823 002122  
824 002122 000120  
825 002124  
826 002124 013176  
827 002126 013320  
828 002130 013366  
829 002132 013576  
830 002134 013746  
831 002136 014112  
832 002140 014252  
833 002142 014412  
834 002144 014564  
835 002146 014770  
836 002150 015174  
837 002152 015376  
838 002154 015562  
839 002156 015674  
840 002160 016134  
841 002162 016430  
842 002164 016670  
843 002166 017130  
844 002170 017370  
845 002172 017630  
846 002174 020120  
847 002176 020422  
848 002200 020662  
849 002202 021122  
850 002204 021362  
851 002206 021622  
852 002210 022062  
853 002212 022322  
854 002214 022562  
855 002216 023022  
856 002220 023330  
857 002222 023726  
858 002224 024412  
859 002226 024722  
860 002230 025262  
861 002232 025570  
862 002234 025736  
863 002236 026100  
864 002240 026252  
865 002242 026456  
866 002244 026632  
867 002246 027006  
868 002250 027156  
869 002252 027350  
870 002254 027570  
871 002256 030016

DISPATCH 80.  
.WORD 80  
LSDISPATCH::  
.WORD T1  
.WORD T2  
.WORD T3  
.WORD T4  
.WORD T5  
.WORD T6  
.WORD T7  
.WORD T8  
.WORD T9  
.WORD T10  
.WORD T11  
.WORD T12  
.WORD T13  
.WORD T14  
.WORD T15  
.WORD T16  
.WORD T17  
.WORD T18  
.WORD T19  
.WORD T20  
.WORD T21  
.WORD T22  
.WORD T23  
.WORD T24  
.WORD T25  
.WORD T26  
.WORD T27  
.WORD T28  
.WORD T29  
.WORD T30  
.WORD T31  
.WORD T32  
.WORD T33  
.WORD T34  
.WORD T35  
.WORD T36  
.WORD T37  
.WORD T38  
.WORD T39  
.WORD T40  
.WORD T41  
.WORD T42  
.WORD T43  
.WORD T44  
.WORD T45  
.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				
907				
908				
909				
910				
911				
912				

:LNT.ED DIFINED AT END OF PROGRAM TO BE LAST TEST NUMBER.

913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942

.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.

BGNHW DFPTBL  
.WORD L10000-LSHW/2  
LSHW::  
DFPTBL::

.WORD 6 :MICRO CPU TYPE  
.WORD 174100 :M8200,4,6,7 CSR ADDRESS  
.WORD 300 :M8200,4,6,7 VECTOR ADDRESS  
.WORD 5000 :INTERRUPT PRIORITY LEVEL  
.WORD 3 :LINE UNIT TYPE  
.WORD 56 :SWITCH PACK #1 (DDCMP LINE #)  
.WORD 0 :SWITCH PACK #2 (BM873 BOOT ADDRESS)  
.WORD 0 :SWITCH PACK #3  
.WORD 0 :TEST CONNECTOR INSTALLED FLAG

ENDHW  
L10000:



943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963

002410  
002410 000000  
002412  
002412  
  
002412  
002412

```
.SBTTL SOFTWARE P-TABLE  
:////  
:// THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM  
:// PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.  
:////  
          BGNSW  SFPTBL  
          .WORD  L10001-LSSW/2  
LSSW::  
SFPTBL::  
  
          ENDSW  
L10001:
```

964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019

002412

.SBTTL GLOBAL EQUATES SECTION

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

EQUALS

; BIT DEFINITIONS

BIT15== 100000  
BIT14== 40000  
BIT13== 20000  
BIT12== 10000  
BIT11== 4000  
BIT10== 2000  
BIT09== 1000  
BIT08== 400  
BIT07== 200  
BIT06== 100  
BIT05== 40  
BIT04== 20  
BIT03== 10  
BIT02== 4  
BIT01== 2  
BIT00== 1

BIT9== BIT09  
BIT8== BIT08  
BIT7== BIT07  
BIT6== BIT06  
BIT5== BIT05  
BIT4== BIT04  
BIT3== BIT03  
BIT2== BIT02  
BIT1== BIT01  
BIT0== BIT00

; EVENT FLAG DEFINITIONS

EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

EF.START== 32.  
EF.RESTART== 31.

; START COMMAND WAS ISSUED  
; RESTART COMMAND WAS ISSUED

1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069

000036  
000035  
000034  
  
000340  
000300  
000240  
000200  
000140  
000100  
000040  
J00000  
  
000004  
000010  
000020  
000040  
000100  
000200  
000400  
001000  
002000  
004000  
010000  
020000  
040000  
100000

EF.CONTINUE== 30.  
EF.NEW== 29.  
EF.PWR== 28.

;; PRIORITY LEVEL DEFINITIONS

PRI07== 340  
PRI06== 300  
PRI05== 240  
PRI04== 200  
PRI03== 140  
PRI02== 100  
PRI01== 40  
PRI00== 0

;; OPERATOR FLAG BITS

EVL== 4  
LOT== 10  
ADR== 20  
IDU== 40  
ISR== 100  
UAM== 200  
BOE== 400  
PNT== 1000  
PRI== 2000  
IXE== 4000  
IBE== 10000  
IER== 20000  
LOE== 40000  
HOE== 100000

: CONTINUE COMMAND WAS ISSUED  
: A NEW PASS HAS BEEN STARTED  
: A POWER-FAIL/POWER-UP OCCURRED

\*\*\*\*\*  
;\* INSTRUCTION DEFINITIONS  
\*\*\*\*\*  
POP2SP=22626 ;INCREMENT STACK TWICE

\*\*\*\*\*  
;\* PROGRAM EVENT FLAG DEFINITIONS  
\*\*\*\*\*

1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080 002412 000000  
1081 002414 000000  
1082  
1083  
1084  
1085  
1086 002416 000000  
1087 002460 002460  
1088 002460 000000  
1089 002522 002522  
1090  
1091  
1092  
1093  
1094 002522 000000  
1095 002524 000000  
1096 002526 000000  
1097 002530 000000  
1098 002532 000000  
1099 002534 000000  
1100 002536 000000  
1101 002540 000000  
1102 002542 000000  
1103 002544 000000  
1104 002546 000000  
1105 002550 000000  
1106 002552 000000  
1107 002554 000001  
1108 002556 041732  
1109 002560 000000  
1110 002560 000000  
1111 002562 000001  
1112 002564 000001  
1113 002566 000001  
1114 002570 000001  
1115 002572 000000  
1116 002574 000000  
1117 002576 000000  
1118 002600 000006  
1119 002602 000000  
1120 002604 000000  
1121 002606 000000  
1122 002610 000000  
1123 002612 000000  
1124 002614 000000  
1125 002616 000000

```
.SBTTL GLOBAL DATA SECTION
:
://////
: THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
: IN MORE THAN ONE TEST.
://////
:*****
:* PROGRAM CONTROL PARAMETERS
:*****
NEXT: .WORD 0 ;ADDRESS OF NEXT TEST TO BE EXECUTED
LOCK: .WORD 0 ;ADDRESS FOR LOCK CURRENT DATA
:*****
:* BUFFERS FOR INPUT-OUTPUT
:*****
TEMP: 0
.=.+40
MDATA: 0
.=.+40
:*****
:* MISCELLANEOUS STORAGE
:*****
$TMP0: .WORD 0 ;SCRATCH STORAGE
LOGDEV: .WORD 0 ;LOGICAL DEVICE NUMBER
PSTACK: .WORD 0 ;BASE LEVEL PROGRAM STACK POINTER
SUBRPC: .WORD 0 ;PC OF SUBR CALL FOR ERROR REPORTS
ERRFLG: .WORD 0 ;SUBROUTINE ERROR FLAG
RETADR: .WORD 0 ;SUBR ERROR RETURN ADDRESS
STRTSW: .WORD 0 ;SWITCHES AT START OF PROGRAM
STAT: .WORD 0 ;M8200,4,6,7 STATUS WORD STORAGE
CLKX: .WORD 0
MASKX: .WORD 0
SAVSP: .WORD 0 ;STACK POINTER STORAGE
SAVPC: .WORD 0 ;PROGRAM COUNTER STORAGE
ZERO: .WORD 0
ONE: .WORD 1
MEMLIM: .WORD MEMEND ;HIGHEST LOCATION FOR NPR'S
MEMSZ: .WORD 0
KMACTV: .BLKW 1 ;M8200,4,6,7 SELECTED ACTIVE
KMINUM: .BLKW 1 ;OCTAL NUMBER OF M8200,4,6,7
SAVACT: .BLKW 1 ;ORIGINAL ACTIVE DEVICES
SAVNUM: .BLKW 1 ;WORKABLE NUMBER
FLAG: .WORD 0 ;SCRATCH STORAGE
RUN: .WORD 0 ;POINTER TO RUNNING DEVICES
MRO: .WORD 0
WTYPE: .WORD 6
TYPE: .WORD 0
PONE: .WORD 0 ; FIRST PASS SWITCH
$GDADR: .WORD 0 ;CONTAINS ADDRESS OF 'GOOD' DATA
$BDADR: .WORD 0 ;CONTAINS ADDRESS OF 'BAD' DATA
$GDDAT: .WORD 0 ;CONTAINS 'GOOD' DATA
$BDDAT: .WORD 0 ;CONTAINS 'BAD' DATA
.WORD 0 ;RESERVED--NOT TO BE USED
```

1126 002620 000000  
1127 002622 000000  
1128 002624 000000  
1129 002626 000000  
1130 002630 000000  
1131 002632 000000  
1132 002634 000000  
1133 002636 000000  
1134 002640 000000

.WORD 0  
FTIME: .WORD 0  
SAVE4: .WORD 0  
SAVE6: .WORD 0  
SCLK: .WORD 0  
SIBS10: .WORD 0  
SIBS11: .WORD 0  
SIBS12: .WORD 0  
SIBS13: .WORD 0

1135  
1136  
1137  
1138  
1139 002642 000 377 000  
1140 002645 377 125 252  
1141 002650 125 252  
1142 002652 000 000 377  
1143 002655 377 125 125  
1144 002660 252 252

\*\*\*\*\*  
;\* DATA PATTERNS  
\*\*\*\*\*

MEMDAT: .BYTE 0,-1,0,-1,125,252,125,252

SPDAT: .BYTE 0,0,-1,-1,125,125,252,252

.EVEN

1145  
1146  
1147  
1148  
1149  
1150 002662 000  
1151 002664 002664  
1152 002664 000  
1153 002665 000

\*\*\*\*\*  
;\* PROGRAM CONTROL FLAGS  
\*\*\*\*\*

INIFLG: .BYTE 0 ;PROGRAM INITIALIZING FLAG

.EVEN

LOKFLG: .BYTE 0 ;LOCK ON CURRENT TEST FLAG

QV.FLG: .BYTE 0 ;QUICK VERIFY FLAG

.EVEN

1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173

\*\*\*\*\*  
;\* DEFINITION OF M8200,4,6,7 STATUS WORDS - STAT1,STAT2,STAT3  
\*\*\*\*\*

STAT1 - BITS 00-08 IS M8200,4,6,7 VECTOR ADDRESS  
BIT15=1 LINE UNIT IS AN M8203  
BIT14=0 NO TEST CONNECTOR(S) USED  
BIT14=1 H-XXX TEST CONNECTOR WILL BE USED  
BIT13=0 LINE UNIT IS AN M8201  
BIT13=1 LINE UNIT IS AN M8202  
BIT12=1 NO LINE UNIT  
BITS 09-11 IS M8200,4,6,7 PRIORITY LEVEL

STAT2 - LOW BYTE IS SWITCH PACK #1 (DDCMP LINE NUMBER)  
HIGH BYTE IS SWITCH PACK #2 (BM873 BOOT ADDRESS)

STAT3 - BIT0=1 DO FREE RUNNING TESTS ON M8200,4,6,7

1174  
1175 002666 000000  
1176 002670 000000  
1177 002672 000000  
1178  
1179  
1180  
1181

\*\*\*\*\*  
STAT1: .WORD 0  
STAT2: .WORD 0  
STAT3: .WORD 0  
\*\*\*\*\*

\*\*\*\*\*  
;\* POINTERS TO M8200,4,6,7 VECTORS AND REGISTERS  
\*\*\*\*\*



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				

1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239

003116				
003116				
003116	034115	030062	000066	
003124				
003124				
003124	055103	046513	040502	
003132	020060	046513	030503	
003140	026461	020102	052123	
003146	052101	041511	050040	
003154	051101	030524	000	
	003162			

```
.SBTTL GLOBAL TEXT SECTION
:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
:% THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
:% MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
:% MORE THAN ONE TEST.
:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
:*****
:* NAMES OF DEVICES SUPPORTED BY PROGRAM
:*****
LSDVTYP:: DEVTYP <M8206>
        .ASCIZ /M8206/
        .EVEN
L$DESC:: DESCRIPT      <CZK MBA0 KMC11-B STATIC PART1>
        .ASCIZ /CZK MBA0 KMC11-B STATIC PART1/
        .EVEN
:
: FORMAT STATEMENTS USED IN PRINT CALLS
:
```

1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295

.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

.MSTCLR:

CLRB @KMCSRH :\*: CLEAR RUN, IF UP

MOVB #BIT6,@KMCSRH :SET INST.

BICB #BIT6!BIT7,@KMCSRH

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.

```
1296  
1297  
1298 003204  
1299 003204 010537 003224  
1300 003210 013701 002704  
1301 003214 000240  
1302 003216 000240  
1303 003220 000240  
1304 003222 000205  
1305 003224 000000  
1306 003226 000024  
1307  
1308  
1309  
1310 003276  
1311  
1312  
1313  
1314  
1315 003276  
1316 003276 152777 000002 177402  
1317 003304 012577 177404  
1318 003310 152777 000003 177370  
1319 003316 142777 000007 177362  
1320 003324 000205  
1321  
1322 003326  
1323  
1324 003326  
1325 003332 000400  
1326 003334  
1327 003340 063220  
1328 003342  
1329 003346 060400  
1330 003350 000207  
1331  
1332 003352  
1333  
1334 003352  
1335 003356 000401  
1336 003360 000207  
1337  
1338 003362  
1339  
1340  
1341 003362  
1342 003366 000402  
1343 003370 000207  
1344  
1345 003372  
1346  
1347  
1348 003372  
1349 003376 000420  
1350 003400 000207  
1351
```

```
CALL=JSR R5,DEBUG  
:  
:  
DEBUG: MOV R5,TESTAD ;SAVE ADDR. OF TEST.  
MOV KMCSR,R1 ;LOAD KMCSR INTP R1.  
NOP  
NOP  
NOP  
TESTAD: RTS R5  
.WORD 0 ;LAST TEST ADDR.  
.BLKW 20. ;PATCH AREA.  
  
ENDBUG:  
; UNSAFE TO PATCH ANY OTHER AREA.  
  
.ROMCLK:  
BISB #BIT1,@KMCSRH  
MOV (R5)+,@KMP06  
BISB #BIT1!BIT0,@KMCSRH  
BICB #BIT2!BIT1!BIT0,@KMCSRH  
RTS R5  
  
CLRALL: ;CLEARS C & Z BITS AND BR  
ROMCLK  
400 ;0 TO BR  
ROMCLK  
63220 ;SP(0) TO BR  
ROMCLK  
60400 ;BR,SP(0) + BR  
RTS PC  
  
SETBR0: ;SETS BR0 BIT  
ROMCLK  
401 ;1 TO BR  
RTS PC  
  
SETBR1: ;THIS SUBROUTINE SETS BR1 BIT  
ROMCLK ;NEXT WORD IS INSTRUCTION  
000402 ;BR_002  
RTS PC  
  
SETBR4: ;THIS SUBROUTINE SETS BR4 BIT  
ROMCLK ;NEXT WORD IS INSTRUCTION  
420  
RTS PC
```

```

1352 003402
1353
1354
1355 003402
1356 003406 000600
1357 003410 000207
1358
1359
1360 003412
1361
1362
1363 003412
1364 003416 000777
1365 003420 000207
1366
1367 003422
1368
1369
1370
1371 003422 017605 000000
1372 003426 062716 000002
1373 003432 005011
1374 003434 052711 000400
1375
1376 003440 005011
1377 003442
1378 003446 061225
1379 003450 116104 000005
1380 003454 000207
1381
1382
1383 003456
1384
1385
1386
1387
1388
1389
1390
1391 003456 005000
1392 003460 012711 002000
1393 003464 010061 000004
1394 003470 012761 000437 000006
1395 003476 052711 020000
1396 003502 005200
1397 003504 022700 002000
1398 003510 001363
1399 003512 005000
1400 003514 012711 002000
1401 003520 016061 003554 000004
1402 003526 016061 003570 000006
1403 003534 052711 020000
1404 003540 005720
1405 003542 022700 000014
1406 003546 001362
1407 003550 005011

SETBR7:
;THIS SUBROUTINE SETS BR7 BIT
ROMCLK ;NEXT WORD IS INSTRUCTION
600
RTS PC

SETZ:
;THIS SUBROUTINE SETS THE Z BIT
ROMCLK ;NEXT WORD IS INSTRUCTION
000777 ;BR_377
RTS PC

RAMDAT:
;THIS SUBROUTINE LOADS R4 WITH THE LOWEST
;8 BITS OF THE CRAM PC.
MOV @ (SP),R5 ;GOOD DATA
ADD #2,(SP) ;ADJUST STACK
CLR (R1) ;CLEAR BIT10
BIS #BIT8,(R1) ;CLOCK INSTRUCTION IN CRAM THAT
;JUMPED TO, IT LOADS BR WITH IT
CLR (R1) ;CLR BIT8
ROMCLK ;NEXT WORD IS INSTRUCTION
061225 ;MOV BR TO PORT 5
MOVB 5(R1),R4 ;PUT 'FOUND' IN R4
RTS PC ;RETURN

MEMSET:
;THIS SUBROUTINE LOADS CRAM WITH SPECIAL INSTRUCTIONS
;FOR THE CRAM JUMP TEST. ALL CRAM LOCATIONS ARE LOADED
;WITH INSTRUCTIONS THAT MOVE A 37 TO THE BR, EXCEPT THE
;FOLLOWING CRAM ADDRESSES: 0,1,4,7,525,1777. THESE LOCATIONS
;CONTAIN INSTRUCTIONS WHICH LOAD THE BR WITH THE LOWEST
;8 BITS OF THAT CRAM ADDRESS.

1$: CLR R0 ;R0 = CRAM ADDRESS
MOV #BIT10,(R1) ;SET ROMO
MOV R0,4(R1) ;LOAD CRAM ADDRESS
MOV #437,6(R1) ;LOAD INSTRUCTION
BIS #BIT13,(R1) ;WRITE INSTRUCTION IN CRAM
INC R0 ;NEXT ADDRESS
CMP #2000,R0 ;DONE YET?
BNE 1$ ;BR IF NO
CLR R0 ;INDEX REGISTER

2$: MOV #BIT10,(R1) ;SET ROMO
MOV CRAM(R0),4(R1) ;LOAD CRAM ADDRESS IN SEL4
MOV INSTU(R0),6(R1) ;LOAD INSTRUCTION TO BE WRITTEN
BIS #BIT13,(R1) ;WRITE CRAM!
TST (R0)+ ;NEXT
CMP #14,R0 ;DONE YET?
BNE 2$ ;BR IF NO
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          MOV      (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
1440 003654 122100          ROMCLK 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      $TMP0,R0    ;GET ADDR.
1456 003706 012704 000010          MOV      #10,R4      ;DO 8 LOADS
1457 003712
1458 003716 010000          ROMCLK 010000          ;MAR < 0
1459 003720          ROMCLK          ;CLR MAR HI
1460 003724 004000
1461 003726 112077 176760          1$: MOV      (R0)+,@KMP04 ;LOAD PORT4
1462 003732          ROMCLK
1463 003736 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		MOV	(R0)+,\$GDDAT		:EXPECTED.
1475	003774	117704	1767*2		MOV	@KMP04,R4		:RECIEVED.
1476	004000	123704	002612		CMP	\$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\$: RTS	PC		:RETURN
1487	004042	000207						
1488								
1489								
1490	004044				SPLD:			
1491								:THIS SUBROUTINE LOADS THE FIRST 8 SCRATCH PAD
1492								:LOCATIONS WITH: 0,0,-1,-1,125,125,252,252
1493								
1494	004044	013600			MOV	@(SP)+,R0		:PUT POINTER TO DATA IN R5
1495	004046	062746	000007		ADD	#2,-(SP)		:ADJUST STACK
1496	004052	005004			CLR	R4		:START AT SP ADDRESS 0
1497	004054	112077	176632		1\$: MOV	(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								:THIS SUBROUTINE CLEARS THE MICRO PROCESSOR C BIT
1510								
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



.SBTTL GLOBAL ERROR REPORT SECTION

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

1543						
1544						
1545						
1546						
1547						
1548						
1549						
1550						
1551						
1552	004210	047045	052045	047045	FM1:	.ASCIZ /%N%T%N/
1553	004216	000				
1554	004217	045	G22516	052501	FMX:	.ASCIZ /%N%AUNIT = %02%A ; FAILING UNIT ADDRESS = %06%N/
1555	004224	044516	020124	020075		
1556	004232	047445	022462	020101		
1557	004240	020073	040506	046111		
1558	004246	047111	020107	047125		
1559	004254	052111	040440	042104		
1560	004262	042522	051523	036440		
1561	004270	022440	033117	047045		
1562	004276	000				
1563	004277	045	031517	051445	TFM1:	.ASCIZ /%03%S5%03%S5%03%S5%03%N2/
1564	004304	022465	031517	051445		
1565	004312	022465	031517	051445		
1566	004320	022465	031517	047045		
1567	004326	000062				
1568	004330	047445	022466	031123	TFM2:	.ASCIZ /%06%S2%06%N2/
1569	004336	047445	022466	031116		
1570	004344	000				
1571	004345	045	031517	051445	TFM5:	.ASCIZ /%03%S5%03%N2/
1572	004352	022465	031517	047045		
1573	004360	000062				
1574	004362	047045	047445	022463	TFM27:	.ASCIZ /%N%03%S5%03%S7%03%N2/
1575	004370	032523	047445	022463		
1576	004376	033523	047445	022463		
1577	004404	031116	000			
1578						
1579						
1580						
1581						
1582	004407	045	030516	040445	FMSG:	.ASCIZ /%N1%A PROGRAM CLOCK = 75 MICROSECONDS%N1/
1583	004414	051120	043517	040522		
1584	004422	020115	046103	041517		
1585	004430	020113	020075	032467		
1586	004436	046440	041511	047522		
1587	004444	042523	047503	042116		
1588	004452	022523	030516	000		
1589	004457	045	030516	040445	SMSG:	.ASCIZ /%N1%A PROGRAM CLOCK = 115 MILLISECONDS%N1/
1590	004464	051120	043517	040522		
1591	004472	020115	046103	041517		
1592	004500	020113	020075	030461		
1593	004506	020065	044515	046114		
1594	004514	051511	041505	047117		
1595	004522	051504	047045	000061		
1596	004530	042522	044507	052123	EM1:	.ASCIZ &REGISTER ADDRESS TEST&
1597	004536	051105	040440	042104		
1598	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	041940	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				



1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806

```
-----  
: MACRO'S NEEDED TO REPORT ERRORS  
-----  
.MACRO MDT0  
.ENDM  
  
.MACRO MDT1  
PRINTB #TFM1,$GDDAT,$BDDAT,$GDADR  
.ENDM  
  
.MACRO MDT2  
PRINTB #TFM2,$GDDAT,$BDDAT  
.ENDM  
  
.MACRO MDT27  
PRINTB #TFM27,MRO,$GDDAT,$BDDAT  
.ENDM  
  
.MACRO SMD,ERNB,ERHM,ERFM  
.LIST  
: ERNB = ERROR NUMBER  
: ERFM = FORMAT NUMBER  
: ERHM = HEADER NUMBER  
.LIST  
BGNMSG ERR'ERNB'  
PRINTB #FMX,LOGDEV,KMCSR  
PRINTB #FM1,#DH'ERHM'  
MDT'ERFM'  
ENDMSG  
  
.ENDM
```

1807	006410		
1808	006410		
1809	006410	013746	002704
1810	006414	013746	002524
1811	006420	012746	004217
1812	006424	012746	000003
1813	006430	010600	
1814	006432	104414	
1815	006434	062706	000010
1816	006440	012746	006257
1817	006444	012746	004210
1818	006450	012746	000002
1819	006454	010600	
1820	006456	104414	
1821	006460	062706	000006
1822	006464	013746	002614
1823	006470	013746	002612
1824	006474	012746	004330
1825	006500	012746	000003
1826	006504	010600	
1827	006506	104414	
1828	006510	062706	000010
1829	006514		
1830	006514	104423	
1831	006516		
1832	006516		
1833	006516	013746	002704
1834	006522	013746	002524
1835	006526	012746	004217
1836	006532	012746	000003
1837	006536	010600	
1838	006540	104414	
1839	006542	062706	000010
1840	006546	012746	006257
1841	006552	012746	004210
1842	006556	012746	000002
1843	006562	010600	
1844	006564	104414	
1845	006566	062706	000006
1846	006572	013746	002614
1847	006576	013746	002612
1848	006602	012746	004330
1849	006606	012746	000003
1850	006612	010600	
1851	006614	104414	
1852	006616	062706	000010
1853	006622		
1854	006622	104423	
1855	006624		
1856	006624		
1857	006624	013746	002704
1858	006630	013746	002524
1859	006634	012746	004217
1860	006640	012746	000003
1861	006644	010600	
1862	006646	104414	

```
ERR1:: SMD 1,2,2
MOV KMCSR,-(SP)
MOV LOGDEV,-(SP)
MOV #FMX,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP
MOV #DH2,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #6,SP
MOV $BDDAT,-(SP)
MOV $GDDAT,-(SP)
MOV #TFM2,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP
L10002: TRAP C$MSG
SMD 2,2,2
ERR2:: MOV KMCSR,-(SP)
MOV LOGDEV,-(SP)
MOV #FMX,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP
MOV #DH2,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #6,SP
MOV $BDDAT,-(SP)
MOV $GDDAT,-(SP)
MOV #TFM2,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP
L10003: TRAP C$MSG
SMD 3,2,2
ERR3:: MOV KMCSR,-(SP)
MOV LOGDEV,-(SP)
MOV #FMX,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
```

1863 006650 062706 000010  
1864 006654 012746 006257  
1865 006660 012746 004210  
1866 006664 012746 000002  
1867 006670 010600  
1868 006672 104414  
1869 006674 062706 000006  
1870 006700 013746 002614  
1871 006704 013746 002612  
1872 006710 012746 004330  
1873 006714 012746 000003  
1874 006720 010600  
1875 006722 104414  
1876 006724 062706 000010  
1877 006730  
1878 006730 104423  
1879 006732  
1880 006732  
1881 006732 013746 002704  
1882 006736 013746 002524  
1883 006742 012746 004217  
1884 006746 012746 000003  
1885 006752 010600  
1886 006754 104414  
1887 006756 062706 000010  
1888 006762 012746 006216  
1889 006766 012746 004210  
1890 006772 012746 000002  
1891 006776 010600  
1892 007000 104414  
1893 007002 062706 000006  
1894 007006 013746 002606  
1895 007012 013746 002614  
1896 007016 013746 002612  
1897 007022 012746 004277  
1898 007026 012746 000004  
1899 007032 010600  
1900 007034 104414  
1901 007036 062706 000012  
1902 007042  
1903 007042 104423  
1904 007044  
1905 007044  
1906 007044 013746 002704  
1907 007050 013746 002524  
1908 007054 012746 004217  
1909 007060 012746 000003  
1910 007064 010600  
1911 007066 104414  
1912 007070 062706 000010  
1913 007074 012746 006216  
1914 007100 012746 004210  
1915 007104 012746 000002  
1916 007110 010600  
1917 007112 104414  
1918 007114 062706 000006

ADD #10,SP  
MOV #DH2,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV \$BDDAT,-(SP)  
MOV \$GDDAT,-(SP)  
MOV #TFM2,-(SP)  
MOV #3,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #10,SP  
L10004: TRAP C\$MSG  
\$MD 4.1.1  
ERR4:: MOV KMCSR,-(SP)  
MOV LOGDEV,-(SP)  
MOV #FMX,-(SP)  
MOV #3,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #10,SP  
MOV #DH1,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV \$GDADR,-(SP)  
MOV \$BDDAT,-(SP)  
MOV \$GDDAT,-(SP)  
MOV #TFM1,-(SP)  
MOV #4,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #12,SP  
L10005: TRAP C\$MSG  
\$MD 5.1.1  
ERR5:: MOV KMCSR,-(SP)  
MOV LOGDEV,-(SP)  
MOV #FMX,-(SP)  
MOV #3,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #10,SP  
MOV #DH1,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP

Line No.	Address	Offset	Value	Instruction	Comment
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				
1928	007154	104423		L10006: TRAP	C\$MSG
1929	007156			SMD	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: TRAP	C\$MSG
1953	007266	104423		SMD	7,3,1
1954	007270			ERR7::	
1955	007270				
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						
1977	007400								
1978	007400	104423							
1979	007402								
1980	007402								
1981	007402	013746	002704						
1982	007406	013746	002524						
1983	007412	012746	004217						
1984	007416	012746	000003						
1985	007422	010600							
1986	007424	104414							
1987	007426	062706	000010						
1988	007432	012746	006273						
1989	007436	012746	004210						
1990	007442	012746	000002						
1991	007446	010600							
1992	007450	104414							
1993	007452	062706	000006						
1994	007456	013746	002606						
1995	007462	013746	002614						
1996	007466	013746	002612						
1997	007472	012746	004277						
1998	007476	012746	000004						
1999	007502	010600							
2000	007504	104414							
2001	007506	062706	000012						
2002	007512								
2003	007512	104423							
2004	007514								
2005	007514								
2006	007514	013746	002704						
2007	007520	013746	002524						
2008	007524	012746	004217						
2009	007530	012746	000003						
2010	007534	010600							
2011	007536	104414							
2012	007540	062706	000010						
2013	007544	012746	006257						
2014	007550	012746	004210						
2015	007554	012746	000002						
2016	007560	010600							
2017	007562	104414							
2018	007564	062706	000006						
2019	007570	013746	002614						
2020	007574	013746	002612						
2021	007600	012746	004330						
2022	007604	012746	000003						
2023	007610	010600							
2024	007612	104414							
2025	007614	062706	000010						
2026	007620								
2027	007620	104423							
2028	007622								
2029	007622								
2030	007622	013746	002704						

```

L10010: TRAP CSPNTB
          ADD #12,SP
          TRAP CSMSG
          SMD 10,3,1
ERR10:: MOV KMCSR,-(SP)
          MOV LOGDEV,-(SP)
          MOV #FMX,-(SP)
          MOV #3,-(SP)
          MOV SP,R0
          TRAP CSPNTB
          ADD #10,SP
          MOV #DH3,-(SP)
          MOV #FM1,-(SP)
          MOV #2,-(SP)
          MOV SP,R0
          TRAP CSPNTB
          ADD #6,SP
          MOV $GDADR,-(SP)
          MOV $BDDAT,-(SP)
          MOV $GDDAT,-(SP)
          MOV #TFM1,-(SP)
          MOV #4,-(SP)
          MOV SP,R0
          TRAP CSPNTB
          ADD #12,SP
L10011: TRAP CSMSG
          SMD 11,2,2
ERR11:: MOV KMCSR,-(SP)
          MOV LOGDEV,-(SP)
          MOV #FMX,-(SP)
          MOV #3,-(SP)
          MOV SP,R0
          TRAP CSPNTB
          ADD #10,SP
          MOV #DH2,-(SP)
          MOV #FM1,-(SP)
          MOV #2,-(SP)
          MOV SP,R0
          TRAP CSPNTB
          ADD #6,SP
          MOV $BDDAT,-(SP)
          MOV $GDDAT,-(SP)
          MOV #TFM2,-(SP)
          MOV #3,-(SP)
          MOV SP,R0
          TRAP CSPNTB
          ADD #10,SP
L10012: TRAP CSMSG
          SMD 12,2,2
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				
2051	007726	104423		L10013:	TRAP C\$MSG
2052	007730				SMD 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				
2068	010004	104423		L10014:	TRAP C\$MSG
2069	010006				SMD 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	CSPNTB
2090	010106	062706	000010		ADD	#10,SP
2091	010112			L10015:		
2092	010112	104423			TRAP	C\$MSG
2093	010114				SMD	15.4.27
2094	010114			ERR15::		
2095	010114	013746	002704		MOV	KMCSR,-(SP)
2096	010120	013746	002524		MOV	LOGDEV,-(SP)
2097	010124	012746	004217		MOV	#FMX,-(SP)
2098	010130	012746	000003		MOV	#3,-(SP)
2099	010134	010600			MOV	SP,R0
2100	010136	104414			TRAP	CSPNTB
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	CSPNTB
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	MRO,-(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	CSPNTB
2115	010220	062706	000012		ADD	#12,SP
2116	010224			L10016:		
2117	010224	104423			TRAP	C\$MSG
2118	010226				SMD	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	CSPNTB
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	CSPNTB
2132	010276	062706	000006		ADD	#6,SP
2133	010302			L10017:		
2134	010302	104423			TRAP	C\$MSG
2135	010304				SMD	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	CSPNTB

Address	Offset	Value	Comment
2143	010330	062706	000010
2144	010334	012746	006215
2145	010340	012746	004210
2146	010344	012746	000002
2147	010350	010600	
2148	010352	104414	
2149	010354	062706	000006
2150	010360		
2151	010360	104423	
2152	010362		
2153	010362		
2154	010362	013746	002704
2155	010366	013746	002524
2156	010372	012746	004217
2157	010376	012746	000003
2158	010402	010600	
2159	010404	104414	
2160	010406	062706	000010
2161	010412	012746	006257
2162	010416	012746	004210
2163	010422	012746	000002
2164	010426	010600	
2165	010430	104414	
2166	010432	062706	000006
2167	010436	013746	002614
2168	010442	013746	002612
2169	010446	012746	004330
2170	010452	012746	000003
2171	010456	010600	
2172	010460	104414	
2173	010462	062706	000010
2174	010466		
2175	010466	104423	
2176	010470		
2177	010470		
2178	010470	013746	002704
2179	010474	013746	002524
2180	010500	012746	004217
2181	010504	012746	000003
2182	010510	010600	
2183	010512	104414	
2184	010514	062706	000010
2185	010520	012746	006215
2186	010524	012746	004210
2187	010530	012746	000002
2188	010534	010600	
2189	010536	104414	
2190	010540	062706	000006
2191	010544		
2192	010544	104423	
2193	010546		
2194	010546		
2195	010546	013746	002704
2196	010552	013746	002524
2197	010556	012746	004217
2198	010562	012746	000003

```
ADD #10,SP
MOV #DH0,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #6,SP
L10020:
TRAP C$MSG
SMD 20.2.2
ERR20::
MOV KMCSR,-(SP)
MOV LOGDEV,-(SP)
MOV #FMX,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP
MOV #DH2,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #6,SP
MOV $BDDAT,-(SP)
MOV $GDDAT,-(SP)
MOV #TFM2,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP
L10021:
TRAP C$MSG
SMD 21.0.0
ERR21::
MOV KMCSR,-(SP)
MOV LOGDEV,-(SP)
MOV #FMX,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP
MOV #DH0,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #6,SP
L10022:
TRAP C$MSG
SMD 22.0.0
ERR22::
MOV KMCSR,-(SP)
MOV LOGDEV,-(SP)
MOV #FMX,-(SP)
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				
2209	010622	104423		L10023:	TRAP C\$MSG
2210	010624				SMD 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:	TRAP C\$MSG
2234	010734	104423			SMD 24,0,0
2235	010736			ERR24::	
2236	010736				
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:	TRAP C\$MSG
2251	011012	104423			SMD 25,2,2
2252	011014			ERR25::	
2253	011014				
2254	011014	013746	002704	MOV	KMCSR,-(SP)

Address	Instruction	Operand 1	Operand 2	Operand 3
2255	MOV	LOGDEV,-(SP)		
2256	MOV	#FMX,-(SP)		
2257	MOV	#3,-(SP)		
2258	MOV	SP,R0		
2259	TRAP	C\$PNTB		
2260	ADD	#10,SP		
2261	MOV	#DH2,-(SP)		
2262	MOV	#FM1,-(SP)		
2263	MOV	#2,-(SP)		
2264	MOV	SP,R0		
2265	TRAP	C\$PNTB		
2266	ADD	#6,SP		
2267	MOV	\$BDDAT,-(SP)		
2268	MOV	\$GDDAT,-(SP)		
2269	MOV	#TFM2,-(SP)		
2270	MOV	#3,-(SP)		
2271	MOV	SP,R0		
2272	TRAP	C\$PNTB		
2273	ADD	#10,SP		
2274				
2275	L10026:	TRAP	C\$MSG	
2276	ERR26::	\$MD	26,2,2	
2277				
2278	MOV	KMCSR,-(SP)		
2279	MOV	LOGDEV,-(SP)		
2280	MOV	#FMX,-(SP)		
2281	MOV	#3,-(SP)		
2282	MOV	SP,R0		
2283	TRAP	C\$PNTB		
2284	ADD	#10,SP		
2285	MOV	#DH2,-(SP)		
2286	MOV	#FM1,-(SP)		
2287	MOV	#2,-(SP)		
2288	MOV	SP,R0		
2289	TRAP	C\$PNTB		
2290	ADD	#6,SP		
2291	MOV	\$BDDAT,-(SP)		
2292	MOV	\$GDDAT,-(SP)		
2293	MOV	#TFM2,-(SP)		
2294	MOV	#3,-(SP)		
2295	MOV	SP,R0		
2296	TRAP	C\$PNTB		
2297	ADD	#10,SP		
2298				
2299	L10027:	TRAP	C\$MSG	
2300	ERR27::	\$MD	27,27,27	
2301				
2302	MOV	KMCSR,-(SP)		
2303	MOV	LOGDEV,-(SP)		
2304	MOV	#FMX,-(SP)		
2305	MOV	#3,-(SP)		
2306	MOV	SP,R0		
2307	TRAP	C\$PNTB		
2308	ADD	#10,SP		
2309	MOV	#DH27,-(SP)		
2310	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	MRO,-(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				
2324	011340	104423		L10030: TRAP	C\$MSG
2325	011342			SMD	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: TRAP	C\$MSG
2348	011446	104423		SMD	29.27.27
2349	011450			ERR29::	
2350	011450				
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	MRO,-(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				SMD	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				SMD	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				SMD	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	CSPNTB
2429	012016	062706	000006	ADD	#6,SP
2430	012022				
2431	012022	104423		L10035:	TRAP
2432	012024				C\$MSG
2433	012024			ERR33::	\$MD
2434	012024	013746	002704		33.2.2
2435	012030	013746	002524	MOV	KMCSR,-(SP)
2436	012034	012746	004217	MOV	LOGDEV,-(SP)
2437	012040	012746	000003	MOV	#FMX,-(SP)
2438	012044	010600		MOV	#3,-(SP)
2439	012046	104414		MOV	SP,R0
2440	012050	062706	000010	TRAP	CSPNTB
2441	012054	012746	006257	ADD	#10,SP
2442	012060	012746	004210	MOV	#DH2,-(SP)
2443	012064	012746	000002	MOV	#FM1,-(SP)
2444	012070	010600		MOV	#2,-(SP)
2445	012072	104414		MOV	SP,R0
2446	012074	062706	000006	TRAP	CSPNTB
2447	012100	013746	002614	ADD	#6,SP
2448	012104	013746	002612	MOV	\$BDDAT,-(SP)
2449	012110	012746	004330	MOV	\$GDDAT,-(SP)
2450	012114	012746	000003	MOV	#TFM2,-(SP)
2451	012120	010600		MOV	#3,-(SP)
2452	012122	104414		MOV	SP,R0
2453	012124	062706	000010	TRAP	CSPNTB
2454	012130			ADD	#10,SP
2455	012130	104423		L10036:	TRAP
2456	012132				C\$MSG
2457	012132			ERR34::	\$MD
2458	012132	013746	002704		34.2.2
2459	012136	013746	002524	MOV	KMCSR,-(SP)
2460	012142	012746	004217	MOV	LOGDEV,-(SP)
2461	012146	012746	000003	MOV	#FMX,-(SP)
2462	012152	010600		MOV	#3,-(SP)
2463	012154	104414		MOV	SP,R0
2464	012156	062706	000010	TRAP	CSPNTB
2465	012162	012746	006257	ADD	#10,SP
2466	012166	012746	004210	MOV	#DH2,-(SP)
2467	012172	012746	000002	MOV	#FM1,-(SP)
2468	012176	010600		MOV	#2,-(SP)
2469	012200	104414		MOV	SP,R0
2470	012202	062706	000006	TRAP	CSPNTB
2471	012206	013746	002614	ADD	#6,SP
2472	012212	013746	002612	MOV	\$BDDAT,-(SP)
2473	012216	012746	004330	MOV	\$GDDAT,-(SP)
2474	012222	012746	000003	MOV	#TFM2,-(SP)
2475	012226	010600		MOV	#3,-(SP)
2476	012230	104414		MOV	SP,R0
2477	012232	062706	000010	TRAP	CSPNTB
2478	012236			ADD	#10,SP
				L10037:	

2479	012236	104423	
2480	012240		
2481	012240		
2482	012240	013746	002704
2483	012244	013746	002524
2484	012250	012746	004217
2485	012254	012746	000003
2486	012260	010600	
2487	012262	104414	
2488	012264	062706	000010
2489	012270	012746	006257
2490	012274	012746	004210
2491	012300	012746	000002
2492	012304	010600	
2493	012306	104414	
2494	012310	062706	000006
2495	012314	013746	002614
2496	012320	013746	002612
2497	012324	012746	004330
2498	012330	012746	000003
2499	012334	010600	
2500	012336	104414	
2501	012340	062706	000010
2502	012344		
2503	012344	104423	
2504	012346		
2505	012346		
2506	012346	013746	002704
2507	012352	013746	002524
2508	012356	012746	004217
2509	012362	012746	000003
2510	012366	010600	
2511	012370	104414	
2512	012372	062706	000010
2513	012376	012746	006257
2514	012402	012746	004210
2515	012406	012746	000002
2516	012412	010600	
2517	012414	104414	
2518	012416	062706	000006
2519	012422	013746	002614
2520	012426	013746	002612
2521	012432	012746	004330
2522	012436	012746	000003
2523	012442	010600	
2524	012444	104414	
2525	012446	062706	000010
2526	012452		
2527	012452	104423	

ERR35::	TRAP	C\$MSG
	SMD	35.2.2
	MOV	KMCSR,-(SP)
	MOV	LOGDEV,-(SP)
	MOV	#FMX,-(SP)
	MOV	#3,-(SP)
	MOV	SP,RO
	TRAP	C\$PNTB
	ADD	#10,SP
	MOV	#DH2,-(SP)
	MOV	#FM1,-(SP)
	MOV	#2,-(SP)
	MOV	SP,RO
	TRAP	C\$PNTB
	ADD	#6,SP
	MOV	\$BDDAT,-(SP)
	MOV	\$GDDAT,-(SP)
	MOV	#TFM2,-(SP)
	MOV	#3,-(SP)
	MOV	SP,RO
	TRAP	C\$PN1B
	ADD	#10,SP
L10040:	TRAP	C\$MSG
	SMD	36.2.2
ERR36::	MOV	KMCSR,-(SP)
	MOV	LOGDEV,-(SP)
	MOV	#FMX,-(SP)
	MOV	#3,-(SP)
	MOV	SP,RO
	TRAP	C\$PNTB
	ADD	#10,SP
	MOV	#DH2,-(SP)
	MOV	#FM1,-(SP)
	MOV	#2,-(SP)
	MOV	SP,RO
	TRAP	C\$PNTB
	ADD	#6,SP
	MOV	\$BDDAT,-(SP)
	MOV	\$GDDAT,-(SP)
	MOV	#TFM2,-(SP)
	MOV	#3,-(SP)
	MOV	SP,RO
	TRAP	C\$PNTB
	ADD	#10,SP
L10041:	TRAP	C\$MSG

2528  
2529  
2530  
2531  
2532  
2533  
2534  
2535  
2536 012454  
2537 012454  
2538  
2539  
2540 012454  
2541 012454 000167  
2542 012456 000000  
2543  
2544  
2545 012460  
2546 012460  
2547 012460 104425  
2548  
2549 012462  
2550 012462  
2551 012462 177777  
2552 012464 177777  
2553 012466 177777  
2554 012470  
2555  
2556 012470  
2557 012470  
2558 012470  
2559 012470  
2560 012470 104461  
2561  
2562  
2563

.SBTTL REPORT CODING SECTION

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

LSRPT:: BGNRPT

EXIT RPT  
.WORD JSJMP  
.WORD L10042-2-.

L10042: ENDRPT  
TRAP CSRPT

LSPROT:: BGNPROT  
-1  
-1  
-1  
ENDPROT

LSAUTO:: BGAUTO  
ENDAUTO  
L10044: TRAP CSAUTO

2564  
2565  
2566  
2567  
2568  
2569  
2570  
2571 012472  
2572 012472  
2573  
2574  
2575 012472 012705 003116  
2576  
2577 012476 010637 002526  
2578 012502 005737 002622  
2579 012506 001013  
2580 012510 013737 000004 002624  
2581 012516 013737 000006 002626  
2582 012524 005037 002604  
2583  
2584 012530 012737 000001 002622  
2585 012536 013737 002624 000004  
2586 012544 013737 002626 000006  
2587  
2588 012552  
2589 012552 012700 000040  
2590 012556 104447  
2591 012560  
2592 012560 103003  
2593 012562 005037 002604  
2594 012566 000414  
2595 012570  
2596  
2597 012570  
2598 012570 012700 000035  
2599 012574 104447  
2600 012576  
2601 012576 103410  
2602  
2603 012600  
2604 012600 012700 000036  
2605 012604 104447  
2606 012606  
2607 012606 103566  
2608  
2609  
2610 012610  
2611 012610 012700 000037  
2612 012614 104447  
2613 012616  
2614 012616 103003  
2615  
2616 012620  
2617  
2618 012620 012737 177777 002524  
2619

```
.SBTTL INITIALIZE SECTION
://////
:/ THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
:/ AT THE BEGINNING OF EACH PASS.
://////

      BGNINIT
LS$INIT::

:INITIALIZE SUBROUTINE STACK
      MOV      #SSTACK,R5
:STORE BASE LEVEL PROGRAM STACK POINTER
      MOV      SP,PSTACK
      TST      FTIME
      BNE      1$
      MOV      @#4,SAVE4
      MOV      @#6,SAVE6
      CLR      PONE                ; INIT FLAG

1$:   MOV      #1,FTIME
      MOV      SAVE4,@#4
      MOV      SAVE6,@#6
:SEE IF PROGRAM JUST STARTED, BR IF YES
      READEF  #EF.START
      MOV      #EF.START,R0
      TRAP    CSREFG
      BNCOMPLETE 33$
      BCC     33$
      CLR     PONE                ; RESET SWITCH
      BR     NEWST

33$:
:SEE IF THIS IS A NEW PASS, BR IF YES
      READEF  #EF.NEW
      MOV      #EF.NEW,R0
      TRAP    CSREFG
      BCOMPLETE  NEWST
      BCS     NEWST
:SEE IF PROGRAM WAS JUST CONTINUED
      READEF  #EF.CONTINUE
      MOV      #EF.CONTINUE,R0
      TRAP    CSREFG
      BCOMPLETE  ENDIT
      BCS     ENDIT

:SEE IF PROGRAM JUST RESTARTED, BR IF NOT
      READEF  #EF.RESTART
      MOV      #EF.RESTART,R0
      TRAP    CSREFG
      BNCOMPLETE  GETPRM
      BCC     GETPRM

NEWST:
:RESET LOGICAL DEVICE TO -1
      MOV      #-1,LOGDEV
```



```
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          :
2692 013072 000240          INC      LOGDEV
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$:      MGV     #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   CSINIT
2711
2712          .EVEN
2713
2714
2715
2716
2717
```



2718  
2719  
2720  
2721  
2722  
2723  
2724  
2725  
2726  
2727  
2728  
2729  
2730  
2731  
2732  
2733  
2734  
2735  
2736

013166  
013166  
  
013166  
013166  
013166 104412

```
.SBTTL CLEANUP CODING SECTION  
:////////////////////  
:// THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED  
:// AT THE END OF EACH PASS.  
:////////////////////  
  
          BGNCLN  
L$CLEAN::  
  
          ENDCLN  
L10046:  TRAP  C$CLEAN
```

2737  
2738  
2739  
2740  
2741  
2742  
2743  
2744  
2745  
2746  
2747  
2748  
2749  
2750  
2751  
2752  
2753  
2754  
2755  
2756

013170  
013170  
  
013170  
013170 104433  
013172  
013172  
013172 104453

```
.SBTTL DROP UNIT SECTION  
:////  
:// THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE  
:// TO NO LONGER BE TESTED.  
:////  
          BGNDU  
LSDU::  
;ISSUE UNIBUS RESET TO CLEAN UP  
          BRESET  
          TRAP    CSRESET  
          ENDDU  
L10047:  
          TRAP    CSDU
```

2757  
2758  
2759  
2760  
2761  
2762  
2763  
2764  
2765  
2766  
2767  
2768  
2769  
2770  
2771  
2772  
2773  
2774  
2775

013174  
013174  
013174  
013174  
013174 104452

.SBTTL ADD UNIT SECTION

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

LSAU: : BGNAU  
          ENDAU  
L10050: TRAP CS AU

.SBTTL HARDWARE TESTS

2776  
2777  
2778  
2779  
2780  
2781 013176  
2782  
2783  
2784  
2785 013176  
2786  
2787  
2788 013176  
2789 013176  
2790 013176 013701 002704  
2791 013202 012705 000004  
2792 013206 012737 013244 000004  
2793 013214 012737 000240 000006  
2794 013222 005711  
2795 013224 000240  
2796 013226  
2797 013226 104410  
2798 013230 000066  
2799 013232 062701 000002  
2800 013236 005305  
2801 013240 001370  
2802 013242 000415  
2803 013244 062706 000004  
2804 013250  
2805 013266 104455  
2806 013270 000001  
2807 013272 004530  
2808 013274 006410  
2809  
2810 013276 013737 002624 000004  
2811 013304 013737 002626 000006  
2812 013312  
2813 013312 104410  
2814 013314 000002  
2815 013316  
2816 013316  
2817 013316 104401

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

BGNTST
T1::
MOV      KMCSR,R1      ;R1 CONTAINS BASE M8200,4,6,7 ADDRESS
MOV      #4,R5         ;4 REGISTERS TO BE TESTED
MOV      #2,R4         ;SET UP TIMEOUT TRAP
MOV      #240,R6       ;LEVEL 5
1$:      TST           ;REFERENCE DEVICE REGISTER
        NOP
        ESCAPE TST
        TRAP      C$ESCAPE
        .WORD     L10051-.
        ADD      #2,R1      ;NEXT REGISTER
        DEC      R5         ;DEC REGISTER COUNT
        BNE     1$         ;BR IF NOT LAST REGISTER
        BR      3$
2$:      ADD      #4,SP
        ERROR    1         ;TIME-OUT ERROR
        TRAP     C$ERDF
        .WORD    1
        .WORD    EM1
        .WORD    ERR1
3$:      MOV      SAVE4,4
        MOV      SAVE6,6
        ESCAPE  TST
        TRAP     C$ESCAPE
        .WORD    L10051-.
ENDTST
L10051: TRAP      C$ETST
```

2818 013320  
2819  
2820  
2821 013320  
2822  
2823  
2824 013320  
2825 013320  
2826 013320  
2827 013324 005011  
2328 013326 005037 002612  
2829 013332 011104  
2830 013334 001413  
2831 013336  
2832 013354 104455  
2833 013356 000032  
2834 013360 005743  
2835 013362 011122  
2836 013364  
2837 013364  
2838 013364  
2839 013364 104401

BADHEAD

:\*\*\*\*\*  
:\*\*\*\*\* TEST 2 \*\*\*\*\*  
:\*VERIFY THAT RUN CAN BE CLEARED  
BADHEAD  
:\*\*\*\*\* TEST 2 \*\*\*\*\*

BGNTST  
T2::

MYINT

CLR (R1) ;CLEAR KMCSR  
CLR \$GDDAT ;CLEAR 'EXPECTED'  
MOV (R1),R4 ;PUT KMCSR IN 'FOUND'  
BEQ 1\$ ;BR IF CLEARED  
ERROR 26 ;ERROR KMCSR NOT CLEARED  
TRAP C\$ERDF  
.WORD 26  
.WORD EM26  
.WORD ERR26

1\$:  
ENDTST  
L10052:

TRAP C\$ETST

2840 013366  
2841  
2842  
2843  
2844  
2845  
2846  
2847  
2848  
2849  
2850  
2851  
2852  
2853  
2854  
2855  
2856  
2857 013366  
2858  
2859  
2860 013366  
2861 013366  
2862 013366  
2863  
2864 013372  
2865 013376 012705 000001  
2866 013402  
2867 013402 104404  
2868 013404 005011  
2869 013406 010537 002612  
2870 013412 010511  
2871 013414 011104  
2872 013416 023704 002612  
2873 013422 001413  
2874 013424  
2875 013442 104455  
2876 013444 000033  
2877 013446 005774  
2878 013450 011230  
2879 013452  
2880 013452 104410  
2881 013454 000014  
2882 013456 005721  
2883 013460 005205  
2884 013462 022705 000005  
2885 013466 001346  
2886 013470  
2887 013470  
2888 013470 104405  
2889 013472 013701 002704  
2890 013476 012705 000001  
2891 013502 012737 013512 002414  
2892 013510  
2893 013510 104404  
2894 013512 010537 002612  
2895 013516 011104

```
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
BADHEAD
:***** TEST 3 *****

BGNTST
T3::

MYINT

MSTCLR                                ;R1 CONTAINS BASE M8200,4,6,7 ADDRESS
MOV #1,R5                              ;MASTER CLEAR M8200,4,6,7
BGNSEG                                ;START PATTERN AT 1
TRAP CSBSEG
CLR (R1)                               ;CLEAR REGISTER
MOV R5,$GDDAT                          ;PUT DATA IN 'EXPECTED'
MOV (R1),R4                             ;WRITE M8200,4,6,7 REGISTER WITH PATTERN
CMP $GDDAT,R4                           ;READ M8200,4,6,7 REGISTER INTO 'FOUND'
BEQ 2$                                  ;IS DATA CORRECT
ERROR 2$                                ;BR IS YES
TRAP CSERDF                             ;DATA ERROR
WORD 27
WORD EM27
WORD ERR27
2$: ESCAPE SEG
TRAP C$ESCAPE
WORD 10000$-
TST (R1)+                               ;NEXT REGISTER
INC R5                                  ;INCREMENT DATA PATTERN
CMP #5,R5                               ;LAST REGISTER?
BNE 1$                                  ;BR IF NO
ENDSEG
10000$: TRAP C$ESEG
MOV KMCSR,R1                            ;BASE M8200,4,6,7 ADDRESS TO R1
MOV #1,R5                                ;RESTART PATTERN AT 1
MOV #3$,LOCK                             ;NEW SCOPI
BGNSEG
TRAP CSBSEG
3$: MOV R5,$GDDAT                       ;PUT DATA IN 'EXPECTED'
MOV (R1),R4                             ;READ COMM. MICR-PROCESSOR FAMILY REGISTER INTO 'FOUND'
```

2896	013520	023704	002612		CMP	\$GDDAT,R4		: IS DATA CORRECT
2897	013524	001413			BEO	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		

```

2916 013576
2917
2918
2919
2920
2921 013576
2922
2923
2924 013576
2925 013576
2926 013576
2927 013602 005037 002576
2928 013606 012702 000001
2929 013612
2930 013612 104404
2931 013614 013701 002704
2932 013620 010237 002612
2933 013624 013711 002612
2934 013630 011104
2935 013632 023704 002612
2936 013636 001413
2937 013640
2938 013656 104455
2939 013660 000033
2940 013662 005774
2941 013664 011230
2942 013666
2943 013666 104410
2944 013670 000052
2945 013672 040211
2946 013674 005037 002612
2947 013700 011104
2948 013702 001413
2949 013704
2950 013722 104455
2951 013724 000002
2952 013726 004556
2953 013730 006516
2954 013732
2955 013732 104410
2956 013734 000006
2957 013736 106302
2958 013740 001325
2959 013742
2960 013742
2961 013742 104405
2962 013744
2963 013744
2964 013744 104401

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
CLR MRO ;MASTER CLEAR M8200.4.6.7
MOV #BIT0,R2 ;INDICATE BSELO
BGNSEG
TRAP C$BSEG
1$: MOV KMCSR,R1 ;PUT REGISTER ADDRESS IN R1
MOV R2,$GDDAT
MOV $GDDAT,(R1) ;WRITE BIT 0
MOV (R1),R4 ;READ CONTROL STATUS REGISTER
CMP $GDDAT,R4 ;IS DATA CORRECT
BEQ 2$ ;BR IF YES
ERROR 27 ;DATA ERROR
TRAP C$ERDF
.WORD 27
.WORD EM27
.WORD ERR27
2$: ESCAPE SEG
TRAP C$ESCAPE
.WORD 10000$-
3$: BIC R2,(R1) ;CLEAR BSELO
CLR $GDDAT ;CLEAR 'EXPECTED'
MOV (R1),R4 ;READ CONTROL STATUS REGISTER
BEQ 4$ ;BR IF ZERO
ERROR 2 ;DATA ERROR BSEL NOT CLEARED
TRAP C$ERDF
.WORD 2
.WORD EM2
.WORD ERR2
4$: ESCAPE SEG
TRAP C$ESCAPE
.WORD 10000$-
ASLB R2
BNE 1$
ENDSEG
10000$: TRAP C$ESEG
ENDTST
L10054: TRAP C$ETST

```



```
2965 013746
2966
2967
2968
2969
2970 013746
2971
2972
2973 013746
2974 013746
2975 013746
2976 013752
2977 013752 104404
2978 013754 013701 002704
2979 013760 012737 001000 002612
2980 013766 013711 002612
2981 013772 011104
2982 013774 023704 002612
2983 014000 001413
2984 014002
2985 014020 104455
2986 014022 000033
2987 014024 005774
2988 014026 011230
2989 014030
2990 014030 104410
2991 014032 000002
2992 014034
2993 014034
2994 014034 104405
2995 014036
2996 014036 104404
2997 014040 042711 001000
2998 014044 005037 002612
2999 014050 011104
3000 014052 001416
3001 014054
3002 014072 104455
3003 014074 000033
3004 014076 005743
3005 014100 011122
3006 014102
3007 014102 104410
3008 014104 000002
3009 014106
3010 014106
3011 014106 104405
3012 014110
3013 014110
3014 014110
3015 014110 104401

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

BGNTST
T5::
MSTCLR ;MASTER CLEAR M8200.4,6,7
BGNSEG
TRAP
C$BSEG
MOV KMCSR,R1 ;PUT REGISTER ADDRESS IN R1
MOV #BIT9,$GDDAT ;PUT DATA IN 'EXPECTED'
MOV $GDDAT,(R1) ;WRITE BIT 9
MOV (R1),R4 ;READ CONTROL STATUS REGISTER
CMP $GDDAT,R4 ;IS DATA CORRECT
BEQ 2$ ;BR IF YES
ERROR 27 ;DATA ERROR
TRAP C$ERDF
.WORD 27
.WORD EM27
.WORD ERR27
2$: ESCAPE SEG
TRAP C$ESCAPE
.WORD 10000$-
ENDSEG
10000$: TRAP C$ESEG
BGNSEG
TRAP
C$BSEG
BIC #BIT9,(R1) ;CLEAR BIT 9
CLR $GDDAT ;CLEAR 'EXPECTED'
MOV (R1),R4 ;READ CONTROL STATUS REGISTER
BEQ 4$ ;BR IF ZERO
ERROR 26 ;DATA ERROR BIT9 NOT CLEARED
TRAP C$ERDF
.WORD 26
.WORD EM26
.WORD ERR26
ESCAPE SEG
TRAP C$ESCAPE
.WORD 10001$-
ENDSEG
10001$: TRAP C$ESEG
4$:
ENDTST
L10055: TRAP C$ETST
```

```
3016 014112          BADHEAD
3017                  :***** TEST 6 *****
3018                  :*CONTROL STATUS REGISTER WRITE/READ TEST
3019                  :*SET BIT11, VERIFY BIT11 WAS SET
3020                  :*CLEAR BIT11, VERIFY BIT11 WAS CLEARED
3021 014112          BADHEAD
3022                  :***** TEST 6 *****
3023
3024 014112          BGNTST
3025 014112          T6::
3026 014112          MSTCLR
3027 014116          BGNSEG          :MASTER CLEAR M8200,4,6,7
3028 014116          TRAP          C$BSEG
3029 014120          104404          002704          1$: MOV KMCSR,R1          :PUT REGISTER ADDRESS IN R1
3030 014124          012737          004000          002612 MOV #BIT11,$GDDAT          :PUT DATA IN 'EXPECTED'
3031 014132          013711          002612          MOV $GDDAT,(R1)          :WRITE BIT 11
3032 014136          011104          MOV (R1),R4          :READ CONTROL STATUS REGISTER
3033 014140          023704          002612          CMP $GDDAT,R4          :IS DATA CORRECT
3034 014144          001413          BEQ 2$          :BR IF YES
3035 014146          ERROR 26          :DATA ERROR
3036 014164          104455          TRAP C$ERDF
3037 014166          000032          .WORD 26
3038 014170          005743          .WORD EM26
3039 014172          011122          .WORD ERR26
3040 014174          ESCAPE 2$:
3041 014174          104410          TRAP C$ESCAPE
3042 014176          000002          .WORD 10000$-
3043 014200          ENDSEG
3044 014200          10000$:
3045 014200          104405          TRAP C$ESEG
3046 014202          BGNSEG
3047 014202          104404          TRAP C$BSEG
3048 014204          042711          004000          3$: BIC #BIT11,(R1)          :CLEAR BIT 11
3049 014210          005037          002612          CLR $GDDAT          :CLEAR 'EXPECTED'
3050 014214          011104          MOV (R1),R4          :READ CONTROL STATUS REGISTER
3051 014216          001414          BEQ 4$          :BR IF ZERO
3052 014220          ERROR 26          :DATA ERROR BIT11 NOT CLEARED
3053 014236          104455          TRAP C$ERDF
3054 014240          000032          .WORD 26
3055 014242          005743          .WORD EM26
3056 014244          011122          .WORD ERR26
3057 014246          ENDSEG
3058 014246          10001$:
3059 014246          104405          TRAP C$ESEG
3060 014250          4$:
3061 014250          ENDTST
3062 014250          L10056:
3063 014250          104401          TRAP C$ETST
```

```
3064 014252          BADHEAD
3065                ;***** TEST 7 *****
3066                ;*CONTROL STATUS REGISTER WRITE/READ TEST
3067                ;*SET BIT12, VERIFY BIT12 WAS SET
3068                ;*CLEAR BIT12, VERIFY BIT12 WAS CLEARED
3069 014252          BADHEAD
3070                ;***** TEST 7 *****
3071
3072 014252          BGNTST
3073 014252          T7::
3074 014252
3075 014256          MSTCLR          MASTER CLEAR M8200,4,6,7
3076 014256 104404  BGNSEG
3077 014260 013701 002704 TRAP          C$BSEG
3078 014264 012737 010000 1$: MOV          KMCSR,R1          ;PUT REGISTER ADDRESS IN R1
3079 014272 013711 002612 MOV          #BIT12,$GDDAT ;PUT DATA IN 'EXPECTED'
3080 014276 011104          MOV          $GDDAT,(R1) ;WRITE BIT 12
3081 014300 023704 002612 MOV          (R1),R4          ;READ CONTROL STATUS REGISTER
3082 014304 001413          CMP          $GDDAT,R4          ;IS DATA CORRECT
3083 014306          BEQ          2$          ;BR IF YES
3084 014324 104455  ERROR          26          ;DATA ERROR
3085 014326 000032  TRAP          C$ERDF
3086 014330 005743          .WORD          26
3087 014332 011122          .WORD          EM26
3088 014334          .WORD          ERR26
3089 014334 104410  2$: ESCAPE          SEG
3090 014336 000002  TRAP          C$ESCAPE
3091 014340          .WORD          10000$-.
3092 014340          ENDSEG
3093 014340 104405  10000$: TRAP          C$ESEG
3094 014342          BGNSEG
3095 014342 104404  TRAP          C$BSEG
3096 014344 042711 010000 3$: BIC          #BIT12,(R1) ;CLEAR BIT 12
3097 014350 005037 002612 CLR          $GDDAT          ;CLEAR 'EXPECTED'
3098 014354 011104          MOV          (R1),R4          ;READ CONTROL STATUS REGISTER
3099 014356 001414          BEQ          4$          ;BR IF ZERO
3100 014360          ERROR          26          ;DATA ERROR BIT12 NOT CLEARED
3101 014376 104455  TRAP          C$ERDF
3102 014400 000032          .WORD          26
3103 014402 005743          .WORD          EM26
3104 014404 011122          .WORD          ERR26
3105 014406          ENDSEG
3106 014406          10001$:
3107 014406 104405  TRAP          C$ESEG
3108 014410          4$:
3109 014410          ENDTST
3110 014410          L10057:
3111 014410 104401  TRAP          C$ETST
```

```
3112 014412          BADHEAD
3113                :***** TEST 8 *****
3114                :*CONTROL OUT REGISTER WRITE/READ TEST
3115                :*FLOAT A ONE THROUGH SEL2
3116 014412          BADHEAD
3117                :***** TEST 8 *****
3118
3119 014412          BGNTST
3120 014412          T8::
3121 014412
3122 014416 012737 000002 002576  MSTCLR          ;MASTER CLEAR M8200,4,6,7
3123 014424 012702 000001          MOV #2,MRO
3124 014430          MOV #1,R2
3125 014430 104404          BGNSEG
3126                TRAP C$BSEG
3127 014432 013701 002710          1$: MOV KMCTL,R1          ;PUT REGISTER ADDRESS IN R1
3128 014436 010237 002612          MOV R2,$GDDAT          ;PUT DATA IN 'EXPECTED'
3129 014442 013711 002612          MOV $GDDAT,(R1)          ;WRITE BIT 0
3130 014446 011104          MOV (R1),R4          ;READ CONTROL OUT REGISTER
3131 014450 023704 002612          CMP $GDDAT,R4          ;IS DATA CORRECT
3132 014454 001413          BEQ 2$          ;BR IF YES
3133 014456          ERROR 27          ;DATA ERROR
3134 014474 104455          TRAP C$ERDF
3135 014476 000033          .WORD 27
3136 014500 005774          .WORD EM27
3137 014502 011230          .WORD ERR27
3138 014504          2$: ESCAPE SEG
3139 014504 104410          TRAP C$ESCAPE
3140 014506 000052          .WORD 10000$-
3141 014510 040211          3$: BIC R2,(R1) ;CLEAR BIT
3142 014512 005037 002612          CLR $GDDAT          ;CLEAR 'EXPECTED'
3143 014516 011104          MOV (R1),R4          ;READ CONTROL OUT REGISTER
3144 014520 001413          BEQ 4$          ;BR IF ZERO
3145 014522          ERROR 27          ;DATA ERROR BIT0 NOT CLEARED
3146 014540 104455          TRAP C$ERDF
3147 014542 000033          .WORD 27
3148 014544 005774          .WORD EM27
3149 014546 011230          .WORD ERR27
3150 014550          4$: ESCAPEE SEG
3151 014550 104410          TRAP C$ESCAPE
3152 014552 000006          .WORD 10000$-
3153 014554 006302          ASL R2
3154 014556 001325          BNE 1$
3155 014560          ENDSEG
3156 014560          10000$:
3157 014560 104405          TRAP C$ESEG
3158 014562          ENDTST
3159 014562          L10060:
3160 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,MRO
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 CSBSEG
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 1044_5 TRAP C$ERDF
3186 014650 000033 .WORD 27
3187 014652 005774 .WORD EM27
3188 014654 011230 .WORD ERR27
3189 014656 65$:
3190 014656 104410 ESCAPE SEG
3191 014660 000010 TRAP C$ESCAPE
3192 014662 000241 .WORD 10000$-.
3193 014664 006105 CLC ;CLEAR CARRY
3194 014666 001350 ROL R5 ;SHIFT TO NEXT BIT
3195 014670 BNE 64$ ;BR IF NOT DONE YET?
3196 014670 ENDSEG
3197 014670 104405 10000$:
3198 014672 012705 000001 TRAP C$ESEG
3199 014676 MOV #1,R5 ;START WITH BIT0
3200 014676 104404 BGNSEG
3201 014700 66$:
3202 014700 005105 TRAP CSBSEG
3203 014702 010537 002612 COM R5 ;CHANGE TO A FLOATING ZERO
3204 014706 013711 002612 MOV R5,$GDDAT ;PUT 'EXPECTED' IN $GDDAT
3205 014712 011104 MOV $GDDAT,(R1) ;WRITE PORT4 REGISTER
3206 014714 023704 002612 MOV (R1),R4
3207 014720 001413 CMP $GDDAT,R4 ;COMPARE EXPECTED AND FOUND
3208 014722 BEQ 67$ ;BR IF OK
3209 014740 104455 ERROR 27 ;WRITE/READ ERROR
3210 014742 000033 TRAP C$ERDF
3211 014744 005774 .WORD 27
3212 014746 011230 .WORD EM27
3213 014750 67$:
3214 014750 104410 ESCAPE SEG
3215 014752 000012 TRAP C$ESCAPE
3216 014754 005105 .WORD 10001$-.
COM R5 ;CHANGE BACK TO A FLOATING ONE
```

3217 014756 000241  
3218 014760 006105  
3219 014762 001346  
3220 014764  
3221 014764  
3222 014764 104405  
3223 014766  
3224 014766  
3225 014766 104401

CLC  
ROL R5  
BNE 66\$  
ENDSEG  
10001\$: TRAP CSESEG  
ENDTST  
L10061: TRAP CSETST

:CLEAR CARRY  
:SHIFT TO NEXT BIT  
:BR IF NOT DONE YET?

```

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,MRO
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$ERDF
3250 015054 000033                .WORD 27
3251 015056 005774                .WORD EM27
3252 015060 011230                .WORD ERR27
3253 015062                65$:
3254 015062 104410                ESCAPE SEG
3255 015064 000010                TRAP C$ESCAPE
3256 015066 000241                .WORD 10000$-.
3257 015070 006105                CLC                    ;CLEAR CARRY
3258 015072 001350                ROL R5                 ;SHIFT TO NEXT BIT
3259 015074                BNE 64$                ;BR IF NOT DONE YET?
3260 015074                ENDSEG
3261 015074 104405                10000$:
3262 015076 012705 000001  TRAP C$ESEG
3263 015102                MOV #1,R5              ;START WITH BIT0
3264 015102 104404                BGNSEG
3265 015104                66$:
3266 015104 005105                TRAP C$BSEG
3267                COM R5                ;CHANGE TO A FLOATING ZERO
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$ERDF
3275 015146 000033                .WORD 27
3276 015150 005774                .WORD EM27
3277 015152 011230                .WORD ERR27
3278 015154                67$:
3279 015154 104410                ESCAPE SEG
3280 015156 000012                TRAP C$ESCAPE
3281 015160 005105                .WORD 10001$-.
                                COM R5                ;CHANGE BACK TO A FLOATING ONE
  
```

3282 015162 000241  
3283 015164 006105  
3284 015166 001346  
3285 015170  
3286 015170  
3287 015170 104405  
3288 J15172  
3289 015172  
3290 015172 104401

CLC  
ROL R5  
BNE 66\$ :CLEAR CARRY  
ENDSEG :SHIFT TO NEXT BIT  
:BR IF NOT DONE YET?  
10001\$: TRAP C\$ESEG  
ENDTST  
L10062: TRAP C\$ETST



```
3291 015174
3292
3293
3294
3295
3296 015174
3297
3298
3299 015174
3300 015174
3301 015174
3302 015200
3303 015204 012705 000001
3304 015210
3305 015210 104404
3306 015212 105011
3307 015214 010537 002612
3308 015220 110511
3309 015222 111104
3310 015224 123704 002612
3311 015230 001413
3312 015232
3313 015250 104455
3314 015252 000002
3315 015254 004556
3316 015256 006516
3317 015260
3318 015260 104410
3319 015262 000024
3320 015264 105721
3321 015266 005205
3322 015270 022705 000011
3323 015274 001346
3324 015276 013701 002704
3325 015302 012705 000001
3326 015306
3327 015306
3328 015306 104405
3329 015310
3330 015310 104404
3331 015312 110537 002612
3332 015316 111104
3333 015320 123704 002612
3334 015324 001413
3335 015326
3336 015344 104455
3337 015346 000002
3338 015350 004556
3339 015352 006516
3340 015354
3341 015354 104410
3342 015356 000014
3343 015360 105721
3344 015362 005205
3345 015364 022705 000011
3346 015370 001350

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
MOV #1,R5 :MASTER CLEAR M8200,4,6,7
BGNSEG :START PATTERN AT 1
TRAP C$BSEG
1$: CLRB (R1) :CLEAR REGISTER
MOV R5,$GDDAT :PUT DATA IN 'EXPECTED'
MOVB R5,(R1) :WRITE M8200,4,6,7 REGISTER WITH PATTERN
MOVB (R1),R4 :READ M8200,4,6,7 REGISTER INTO 'FOUND'
CMPB $GDDAT,R4 :IS DATA CORRECT
BEQ 2$ :BR IF YES
ERROR 2 :DATA ERROR
TRAP C$ERDF
.WORD 2
.WORD EM2
.WORD ERR2
2$: ESCAPE SEG
TRAP C$ESCAPE
.WORD 10000$-
TSTB (R1)+ :NEXT REGISTER
INC R5 :INCREMENT DATA PATTERN
CMP #11,R5 :LAST REGISTER?
BNE 1$ :BR IF NO
MOV KMCSR,R1 :BASE M8200,4,6,7 ADDRESS TO R1
MOV #1,R5 :RESTART PATTERN AT 1
ENDSEG
10000$: TRAP C$ESEG
BGNSEG
TRAP C$BSEG
3$: MOVB R5,$GDDAT :PUT DATA IN 'EXPECTED'
MOVB (R1),R4 :READ COMM.MICRO-PROCESSOR FAMILY REGISTER INTO 'FOUND'
CMPB $GDDAT,R4 :IS DATA CORRECT
BEQ 4$ :BR IF YES
ERROR 2 :DUAL ADDRESSING ERROR
TRAP C$ERDF
.WORD 2
.WORD EM2
.WORD ERR2
4$: ESCAPE SEG
TRAP C$ESCAPE
.WORD 10001$-
TSTB (R1)+ :NEXT REGISTER
INC R5 :INCREMENT PATTERN
CMP #11,R5 :LAST REGISTER?
BNE 3$ :BR IF NO
```

3347 015372  
3348 015372  
3349 015372 104405  
3350 015374  
3351 015374  
3352 015374 104401

ENDSEG  
10001S: TRAP CSESEG  
ENDTST  
L10063: TRAP CSETST

```
3353 015376          3ADHEAD
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
3364 015376          MSTCLR                ;R1 CONTAINS BASE MB200,4,6,7 ADDRESS
3365 015402          MYINT                ;MASTER CLEAR MB200,4,6,7
3366 015406          BGNSEG
3367 015406 104404   TRAP
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$ERDF
3376 015460 000032   .WORD  26
3377 015462 005743   .WORD  EM26
3378 015464 011122   .WORD  ERR26
3379 015466          2$:   .WORD  ERR26
3380 015466 104410   ESCAPE  SEG
3381 015470 000002   TRAP  C$ESCAPE
3382 015472          .WORD  10000$-
3383 015472          ENDSEG
3384 015472 104405          10000$:
3385 015474 012737 177777 002612   TRAP  C$ESEG
3386 015502          MOV      #-1,$GDDAT          ;PUT 'EXPECTED' IN $GDDAT
3387 015502 104404   BGNSEG
3388 015504 013761 002612 000006 3$:   TRAP  C$BSEG
3389 015512 016104 000006   MOV      $GDDAT,6(R1)    ;WRITE ALL ONES TO THE IR
3390 015516 023704 002612   MOV      6(R1),R4        ;READ THE IR
3391 015522 001413   CMP      $GDDAT,R4      ;IS IT ALL ONES?
3392 015524          BEQ      4$              ;BR IF YES
3393 015542 104455   ERROR  23          ;ERROR IR IS NOT = ALL ONES
3394 015544 000027   TRAP  C$ERDF
3395 015546 005475   .WORD  23
3396 015550 010624   .WORD  EM23
3397 015552          4$:   .WORD  ERR23
3398 015552 104410   ESCAPE  SEG
3399 015554 000002   TRAP  C$ESCAPE
3400 015556          .WORD  10001$-
3401 015556          ENDSEG
3402 015556 104405          10001$:
3403 015560          TRAP  C$ESEG
3404 015560          ENDTST
3405 015560 104401          L10064:
3405 015560          TRAP  C$ETST
```

3406 015562  
3407 015562  
3408 015562  
3409 015562  
3410 015562  
3411 015562  
3412 015562  
3413 015562  
3414 015562  
3415 015562  
3416 015562  
3417 015562  
3418 015562  
3419 015566  
3420 015572 012761 000377 000004  
3421 015600 012711 001000  
3422 015604 012761 121105 000006  
3423 015612 052711 001400  
3424 015616 000240  
3425 015620 012737 177777 002612  
3426 015626 116104 000005  
3427 015632 123704 002612  
3428 015636 001413  
3429 015640  
3430 015656 104455  
3431 015660 000034  
3432 015662 006023  
3433 015664 011342  
3434  
3435 015666  
3436 015666 104410  
3437 015670 000002  
3438  
3439 015672  
3440 015672  
3441 015672 104401

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

BGNTST  
T13::

MYINT  
MSTCLR  
MOV #377,4(R1) ;PORT4 HI BYTE=1'S  
MOV #BIT9,(R1) ;SET ROMI  
MOV #121105,6(R1) ;INSTR TO PORT 6.  
BIS #BIT8!BIT9,(R1) ;CLK INSTR.  
NOP  
MOV #-1,\$GDDAT ;EXPECT ALL ONES.  
MOVB 5(R1),R4 ;READ FOUND.  
CMPB \$GDDAT,R4 ;DATA CORRECT?  
BEQ 1\$  
ERROR 28  
TRAP C\$ERDF  
.WORD 28  
.WORD EM28  
.WORD ERR28

1\$: ESCAPE TST  
TRAP C\$ESCAPE  
.WORD L10065-

ENDTST  
L10065:  
TRAP C\$ETST

3442	015674				BADHEAD		
3443					:***** TEST 14 *****		
3444					:*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST		
3445					:*FLOAT A 1 THROUGH 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		
3453	015700	012737	000000	002576	MOV #0,MRO		:MASTER CLEAR M8200,4,6,7
3454	015706	012705	000001		MOV #5		:SAVE REGISTER ADDRESS FOR TYPEOUT
3455							:START WITH BIT 0
3456	015712				MYINT		
3457	015716				BGNSEG		
3458	015716	104404			TRAP CSBSEG		
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 IBUS* REGISTER 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 CSERDF		
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 1, 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 CSBSEG		
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  
3499 016106 000033  
3500 016110 005774  
3501 016112 011230  
3502 016114  
3503 016114 104410  
3504 016116 000012  
3505 016120 005105  
3506 016122 000241  
3507 016124 106105  
3508 016126 001337  
3509 016130  
3510 016130  
3511 016130 104405  
3512 016132  
3513 016132  
3514 016132 104401

68\$: TRAP C\$ERDF  
.WORD 27  
.WORD EM27  
.WORD ERR27  
ESCAPE SEG  
TRAP C\$ESCAPE  
.WORD 10001\$-.  
COM R5  
CLC :CHANGE TO FLOATING 1  
ROLB R5 :CLEAR CARRY  
BNE 67\$ :SHIFT BIT IN R5  
ENDSEG :IF R5=0 THEN DONE  
10001\$: TRAP C\$ESEG  
ENDTST  
L10066: TRAP C\$ETST



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



```
3591 016430          BADHEAD
3592                ;***** TEST 16 *****
3593                ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
3594                ;*FLOAT A 1 THROUGH IBUS* REGISTER 2
3595                ;*FLOAT A 0 THROUGH IBUS* REGISTER 2
3596 016430          BADHEAD
3597                ;***** TEST 16 *****
3598
3599 016430          BGNTST
3600 016430          T16::
3601 016430
3602 016434 012737 000002 002576  MSTCLR          ;MASTER CLEAR M8200,4,6,7
3603 016442 012705 000001          MOV #2,MRO      ;SAVE REGISTER ADDRESS FOR TYPEOUT
3604 016446          MOV #1,R5      ;START WITH BIT 0
3605 016452          MYINT
3606 016452 104404          BGNSEG
3607 016454          TRAP C$BSEG
3608 016454 010561 000004          64$: MOV R5,4(R1)      ;PUT PATTERN INTO PORT4
3609 016460          ROMCLK          ;NEXT WORD IS INSTRUCTION, BBN
3610 016464 121102          121100!2      ;MOV DATA TO IBUS* REGISTER 0
3611 016466          ROMCLK          ;NEXT WORD IS INSTRUCTION, BBN
3612 016472 121045          121005!<2*20>      ;READ FROM IBUS* REGISTER 2
3613 016474 010537 002612          MOV R5,$GDDAT      ;PUT EXPECTED IN $GDDAT
3614 016500 116104 000005          MOVB 5(R1),R4      ;PUT 'FOUND' INTO R4
3615 016504 123704 002612          CMPB $GDDAT,R4    ;DATA CORRECT?
3616 016510 001413          BEQ 65$          ;BR IF YES
3617 016512          ERROR 27      ;ERROR
3618 016530 104455          TRAP C$ERDF
3619 016532 000033          .WORD 27
3620 016534 005774          .WORD EM27
3621 016536 011230          .WORD ERR27
3622 016540          65$: ESCAPE SEG
3623 016540 104410          TRAP C$ESCAPE
3624 016542 000010          .WORD 10000$-
3625 016544 000241          CLC          ;CLEAR CARRY
3626 016546 106105          ROLB R5      ;SHIFT BIT IN R5
3627 016550 001341          BNE 64$      ;IF R5=0 THEN DONE
3628 016552          ENDSEG
3629 016552          10000$:
3630 016552 104405          TRAP C$ESEG
3631 016554 012705 000001          MOV #1,R5      ;START WITH BIT 0
3632 016560          BGNSEG
3633 016560 104404          TRAP C$BSEG
3634 016562          67$:
3635 016562 005105          COM R5
3636 016564 010561 000004          MOV R5,4(R1)      ;PUT PATTERN INTO PORT4
3637 016570          ROMCLK          ;NEXT WORD IS INSTRUCTION, BBN
3638 016574 121102          121100!2      ;MOV DATA TO IBUS* REGISTER 2
3639 016576          ROMCLK          ;NEXT WORD IS INSTRUCTION, BBN
3640 016602 121045          121005!<2*20>      ;READ FROM IBUS* REGISTER 2
3641 016604 010537 002612          MOV R5,$GDDAT      ;PUT EXPECTED IN $GDDAT
3642 016610 116104 000005          MOVB 5(R1),R4      ;PUT 'FOUND' INTO R4
3643 016614 123704 002612          CMPB $GDDAT,R4    ;DATA CORRECT?
3644 016620 001413          BEQ 68$          ;BR IF YES
3645 016622          ERROR 27      ;ERROR
3646 016640 104455          TRAP C$ERDF
```

3647 016642 000033  
3648 016644 005774  
3649 016646 011230  
3650 016650  
3651 016650 104410  
3652 016652 000012  
3653 016654 005105  
3654 016656 000241  
3655 016660 106105  
3656 016662 001337  
3657 016664  
3658 016664  
3659 016664 104405  
3660 016666  
3661 016666  
3662 016666 104401

68\$: .WORD 27  
.WORD EM27  
.WORD ERR27  
ESCAPE SEG  
TRAP C\$ESCAPE  
.WORD 10001\$-.  
COM R5  
CLC  
ROLB R5  
BNE 67\$  
ENDSEG  
10001\$: TRAP C\$ESEG  
ENDTST  
L10070: TRAP C\$ETST

:CHANGE TO FLOATING 1  
:CLEAR CARRY  
:SHIFT BIT IN R5  
:IF R5=0 THEN DONE

3663	016670					BADHEAD	
3664						:***** TEST 17 *****	
3665						:*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST	
3666						:*FLOAT A 1 THROUGH IBUS* REGISTER 3	
3667						:*FLOAT A 0 THROUGH IBUS* REGISTER 3	
3668	016670					BADHEAD	
3669						:***** TEST 17 *****	
3670							
3671	016670					BGNTST	
3672	016670					T17::	
3673	016670					MSTCLR	
3674	016674	012737	000003	002576		MOV #3,MRO	:MASTER CLEAR M8200,4,6,7
3675	016702	012705	000001			MOV #1,R5	:SAVE REGISTER ADDRESS FOR TYPEOUT
3676	016706					MYINT	:START WITH BIT 0
3677	016712					BGNSEG	
3678	016712	104404				TRAP CSBSEG	
3679	016714				64\$:		
3680	016714	010561	000004			MOV R5,4(R1)	:PUT PATTERN INTO PORT4
3681	016720					ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
3682	016724	121103				121100!3	:MOV DATA TO IBUS* REGISTER 3
3683	016726					ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
3684	016732	121065				121005!<3*20>	:READ FROM IBUS* REGISTER 3
3685	016734	010537	002612			MOV R5,\$GDDAT	:PUT EXPECTED IN \$GDDAT
3686	016740	116104	000005			MOVB 5(R1),R4	:PUT 'FOUND' INTO R4
3687	016744	123704	002612			CMPB \$GDDAT,R4	:DATA CORRECT?
3688	016750	001413				BEQ 65\$	:BR IF YES
3689	016752					ERROR 27	:ERROR
3690	016770	104455				TRAP C\$ERDF	
3691	016772	000033				.WORD 27	
3692	016774	005774				.WORD EM27	
3693	016776	011230				.WORD ERR27	
3694	017000				65\$:	ESCAPE SEG	
3695	017000	104410				TRAP C\$ESCAPE	
3696	017002	000010				.WORD 10000\$-	
3697	017004	000241				CLC	:CLEAR CARRY
3698	017006	106105				ROLB R5	:SHIFT BIT IN R5
3699	017010	001341				BNE 64\$	:IF R5=0 THEN DONE
3700	017012					ENDSEG	
3701	017012				10000\$:		
3702	017012	104405				TRAP C\$ESEG	
3703	017014	012705	000001			MOV #1,R5	:START WITH BIT 0
3704	017020					BGNSEG	
3705	017020	104404				TRAP CSBSEG	
3706	017022				67\$:		
3707	017022	005105				COM R5	
3708	017024	010561	000004			MOV R5,4(R1)	:PUT PATTERN INTO PORT4
3709	017030					ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
3710	017034	121103				121100!3	:MOV DATA TO IBUS* REGISTER 3
3711	017036					ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
3712	017042	121065				121005!<3*20>	:READ FROM IBUS* REGISTER 3
3713	017044	010537	002612			MOV R5,\$GDDAT	:PUT EXPECTED IN \$GDDAT
3714	017050	116104	000005			MOVB 5(R1),R4	:PUT 'FOUND' INTO R4
3715	017054	123704	002612			CMPB \$GDDAT,R4	:DATA CORRECT?
3716	017060	001413				BEQ 68\$	:BR IF YES
3717	017062					ERROR 27	:ERROR
3718	017100	104455				TRAP C\$ERDF	

3719 017102 000033  
3720 017104 005774  
3721 017106 011230  
3722 017110  
3723 017110 104410  
3724 017112 000012  
3725 017114 005105  
3726 017116 000241  
3727 017120 106105  
3728 017122 001337  
3729 017124  
3730 017124  
3731 017124 104405  
3732 017126  
3733 017126  
3734 017126 104401

68\$: .WORD 27  
.WORD EM27  
.WORD ERR27  
ESCAPE SEG  
TRAP C\$ESCAPE  
.WORD 10001\$-  
COM R5  
CLC ;CHANGE TO FLOATING 1  
ROLB R5 ;CLEAR CARRY  
BNE 67\$ ;SHIFT BIT IN R5  
ENDSEG ;IF R5=0 THEN DONE  
10001\$: TRAP C\$ESEG  
ENDTST  
L10071: TRAP C\$ETST

3735	017130				BADHEAD	
3736					:***** TEST 18 *****	
3737					:*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST	
3738					:*FLOAT A 1 THROUGH IBUS* REGISTER 4	
3739					:*FLOAT A 0 THROUGH IBUS* REGISTER 4	
3740	017130				BADHEAD	
3741					:***** TEST 18 *****	
3742						
3743	017130				BGNTST	
3744	017130				T18::	
3745	017130				MSTCLR	:MASTER CLEAR M8200,4,6,7
3746	017134	012737	000004	002576	MOV #4,MRO	:SAVE REGISTER ADDRESS FOR TYPEOUT
3747	017142	012705	000001		MOV #1,R5	:START WITH BIT 0
3748	017146				MYINT	
3749	017152				BGNSEG	
3750	017152	104404			TRAP C\$BSEG	
3751	017154					
3752	017154	010561	000004	64\$:	MOV R5,4(R1)	:PUT PATTERN INTO PORT4
3753	017160				ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
3754	017164	121104			121100!4	:MOV DATA TO IBUS* REGISTER 4
3755	017166				ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
3756	017172	121105			121005!<4*20>	:READ FROM IBUS* REGISTER 4
3757	017174	010537	002612		MOV R5,\$GDDAT	:PUT EXPECTED IN \$GDDAT
3758	017200	116104	000005		MOVB 5(R1),R4	:PUT 'FOUND' INTO R4
3759	017204	123704	002612		CMPB \$GDDAT,R4	:DATA CORRECT?
3760	017210	001413			BEQ 65\$	:BR IF YES
3761	017212				ERROR 27	:ERROR
3762	017230	104455			TRAP C\$ERDF	
3763	017232	000033			.WORD 27	
3764	017234	005774			.WORD EM27	
3765	017236	011230			.WORD ERR27	
3766	017240			65\$:	ESCAPE SEG	
3767	017240	104410			TRAP C\$ESCAPE	
3768	017242	000010			.WORD 10000\$-	
3769	017244	000241			CLC	:CLEAR CARRY
3770	017246	106105			ROLB R5	:SHIFT BIT IN R5
3771	017250	001341			BNE 64\$	:IF R5=0 THEN DONE
3772	017252				ENDSEG	
3773	017252			10000\$:		
3774	017252	104405			TRAP C\$ESEG	
3775	017254	012705	000001		MOV #1,R5	:START WITH BIT 0
3776	017260				BGNSEG	
3777	017260	104404			TRAP C\$BSEG	
3778	017262			67\$:		
3779	017262	005105			COM R5	
3780	017264	010561	000004		MOV R5,4(R1)	:PUT PATTERN INTO PORT4
3781	017270				ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
3782	017274	121104			121100!4	:MOV DATA TO IBUS* REGISTER 4
3783	017276				ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
3784	017302	121105			121005!<4*20>	:READ FROM IBUS* REGISTER 4
3785	017304	010537	002612		MOV R5,\$GDDAT	:PUT EXPECTED IN \$GDDAT
3786	017310	116104	000005		MOVB 5(R1),R4	:PUT 'FOUND' INTO R4
3787	017314	123704	002612		CMPB \$GDDAT,R4	:DATA CORRECT?
3788	017320	001413			BEQ 68\$	:BR IF YES
3789	017322				ERROR 27	:ERROR
3790	017340	104455			TRAP C\$ERDF	

3791 017342 000033  
3792 017344 005774  
3793 017346 011230  
3794 017350  
3795 017350 104410  
3796 017352 000012  
3797 017354 005105  
3798 017356 000241  
3799 017360 106105  
3800 017362 001337  
3801 017364  
3802 017364  
3803 017364 104405  
3804 017366  
3805 017366  
3806 017366 104401

68\$: .WORD 27  
.WORD EM27  
.WORD ERR27  
ESCAPE SEG  
TRAP C\$ESCAPE  
.WORD 10001\$-  
COM R5  
CLC  
ROLB R5  
BNE 67\$  
ENDSEG  
10001\$: TRAP C\$ESEG  
ENDTST  
L10072: TRAP C\$ETST

:CHANGE TO FLOATING 1  
:CLEAR CARRY  
:SHIFT BIT IN R5  
:IF R5=0 THEN DONE

3807	017370				BADHEAD	
3808					:***** TEST 19 *****	
3809					:*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST	
3810					:*FLOAT A 1 THROUGH IBUS* REGISTER 5	
3811					:*FLOAT A 0 THROUGH IBUS* REGISTER 5	
3812	017370				BADHEAD	
3813					:***** TEST 19 *****	
3814						
3815	017370					
3816	017370				BGNTST	
3817	017370				T19::	
3818	017374	012737	000005	002576	MSTCLR	:MASTER CLEAR M8200,4,6,7
3819	017402	012705	000001		MOV #5,MRO	:SAVE REGISTER ADDRESS FOR TYPEOUT
3820	017406				MOV #1,R5	:START WITH BIT 0
3821	017412				MYINT	
3822	017412	104404			BGNSEG	
3823	017414				TRAP CSBSEG	
3824	017414	010561	000004		64\$:	
3825	017420				MOV R5,4(R1)	:PUT PATTERN INTO PORT4
3826	017424	121105			ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
3827	017426				121100!5	:MOV DATA TO IBUS* REGISTER 5
3828	017432	121125			ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
3829	017434	010537	002612		121005!<5*20>	:READ FROM IBUS* REGISTER 5
3830	017440	116104	000005		MOV R5,\$GDDAT	:PUT EXPECTED IN \$GDDAT
3831	017444	123704	002612		MOVB 5(R1),R4	:PUT 'FOUND' INTO R4
3832	017450	001413			CMPB \$GDDAT,R4	:DATA CORRECT?
3833	017452				BEO 65\$	:BR IF YES
3834	017470	104455			ERROR 27	:ERROR
3835	017472	000033			TRAP CSERDF	
3836	017474	005774			.WORD 27	
3837	017476	011230			.WORD EM27	
3838	017500				.WORD ERR27	
3839	017500	104410			65\$:	
3840	017502	000010			ESCAPE SEG	
3841	017504	000241			TRAP C\$ESCAPE	
3842	017506	106105			.WORD 10000\$-	
3843	017510	001341			CLC	:CLEAR CARRY
3844	017512				ROLB R5	:SHIFT BIT IN R5
3845	017512				BNE 64\$	:IF R5=0 THEN DONE
3846	017512	104405			ENDSEG	
3847	017514	012705	000001		10000\$:	
3848	017520				TRAP C\$ESEG	
3849	017520	104404			MOV #1,R5	:START WITH BIT 0
3850	017522				BGNSEG	
3851	017522	005105			TRAP CSBSEG	
3852	017524	010561	000004		67\$:	
3853	017530				COM R5	
3854	017534	121105			MOV R5,4(R1)	:PUT PATTERN INTO PORT4
3855	017536				ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
3856	017542	121125			121100!5	:MOV DATA TO IBUS* REGISTER 5
3857	017544	010537	002612		ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
3858	017550	116104	000005		121005!<5*20>	:READ FROM IBUS* REGISTER 5
3859	017554	123704	002612		MOV R5,\$GDDAT	:PUT EXPECTED IN \$GDDAT
3860	017560	001413			MOVB 5(R1),R4	:PUT 'FOUND' INTO R4
3861	017562				CMPB \$GDDAT,R4	:DATA CORRECT?
3862	017600	104455			BEO 68\$	:BR IF YES
					ERROR 27	:ERROR
					TRAP CSERDF	

3863 017602 000033  
3864 017604 005774  
3865 017606 011230  
3866 017610  
3867 017610 104410  
3868 017612 000012  
3869 017614 005105  
3870 017616 000241  
3871 017620 106105  
3872 017622 001337  
3873 017624  
3874 017624  
3875 017624 104405  
3876 017626  
3877 017626  
3878 017626 104401

68\$: .WORD 27  
.WORD EM27  
.WORD ERR27  
ESCAPE SEG  
TRAP C\$ESCAPE  
.WORD 10001\$-  
COM R5  
CLC  
ROLB R5  
BNE 67\$  
ENDSEG  
10001\$: TRAP C\$ESEG  
ENDTST  
L10073: TRAP C\$ETST

:CHANGE TO FLOATING 1  
:CLEAR CARRY  
:SHIFT BIT IN R5  
:IF R5=0 THEN DONE



```
3879 017630 BADHEAD
3880 :***** TEST 20 *****
3881 :*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
3882 :*FLOAT A 1 THROUGH 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
3890 017534 012737 000010 002576 MSTCLR ;MASTER CLEAR M8200,4,6,7
3891 017642 012705 000001 MOV #10,MRO ;SAVE REGISTER ADDRESS FOR TYPEOUT
3892 017646 MOV #1,R5 ;START WITH BIT 0
3893 017652 MYINT
3894 017652 104404 BGNSEG
3895 017654 TRAP CSBSEG
3896 017654 010561 000004 64$: MOV R5,4(R1) ;PUT PATTERN INTO PORT4
3897 017660 042761 000141 000004 BIC #141,4(R1) ;CLEAR UNWANTED BITS
3898 017666 ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
3899 017672 121110 121100!10 ;MOV DATA TO IBUS* REGISTER 10
3900 017674 ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
3901 017700 121205 121005!<10*20> ;READ FROM IBUS* REGISTER 10
3902 017702 010537 002612 MOV R5,$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 CSERDF
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 CSBSEG
3924 017776 67$: COM R5
3925 017776 005105 MOV R5,4(R1) ;PUT PATTERN INTO PORT4
3926 020000 010561 000004 BIC #141,4(R1) ;CLEAR UNWANTED BITS
3927 020004 042761 000141 000004 ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
3928 020012 121110 121100!10 ;MOV DATA TO IBUS* REGISTER 10
3929 020016 121110 ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
3930 020020 121205 121005!<10*20> ;READ FROM IBUS* REGISTER 10
3931 020024 121205 MOV R5,$GDDAT ;PUT EXPECTED IN $GDDAT
3932 020026 010537 002612 BIC #141,$GDDAT ;CLEAR UNWANTED BITS
3933 020032 042737 000141 002612 MOVB 5(R1),R4 ;PUT 'FOUND' INTO R4
3934 020040 116104 000005
```

3935 020044 123704 002612  
3936 020050 00:415  
3937 020052  
3938 020070 104455  
3939 020072 000033  
3940 020074 005774  
3941 020076 011230  
3942 020100  
3943 020100 104410  
3944 020102 000012  
3945 020104 005105  
3946 020106 000241  
3947 020110 106105  
3948 020112 001331  
3949 020114  
3950 020114  
3951 020114 104405  
3952 020116  
3953 020116  
3954 020116 104401

```
        CMPB    $GDDAT,R4
        BEQ     68$          :DATA CORRECT?
        ERROR   27          :BR IF YES
        TRAP    C$ERDF      :ERROR
        .WORD   27
        .WORD   EM27
        .WORD   ERR27
        ESCAPE  SEG
        TRAP    C$ESCAPE
        .WORD   10001$-
68$:    COM     R5          :CHANGE TO FLOATING 1
        CLC
        ROLB   R5          :CLEAR CARRY
        BNE    67$          :SHIFT BIT IN R5
        ENDSEG
10001$: TRAP    C$ESEG
ENDTST
L10074: TRAP    C$ETST
```

```
3955 020120 BADHEAD
3956 :***** TEST 21 *****
3957 :*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
3958 :*FLOAT A 1 THROUGH IBUS* REGISTER 11
3959 :*FLOAT A 0 THROUGH IBUS* REGISTER 11
3960 020120 BADHEAD
3961 :***** TEST 21 *****
3962
3963 020120 9GNTST
3964 020120 T21::
3965 020120
3966 020124 012737 000011 002576 MSTCLR :MASTER CLEAR MB200,4,6,7
3967 020132 012705 000001 MOV #11,MRO :SAVE REGISTER ADDRESS FOR TYPEOUT
3968 020136 MYINT #1,R5 :START WITH BIT 0
3969 020142 BGNSEG
3970 020142 104404 TRAP CSBSEG
3971 020144
3972 020144 010561 000004 64$: MOV R5,4(R1) :PUT PATTERN INTO PORT4
3973 020150 042761 000222 000004 BIC #222,4(R1) :CLEAR UNWANTED BITS
3974 020156 ROMCLK :NEXT WORD IS INSTRUCTION, BBN
3975 020162 121111 121100!11 :MOV DATA TO IBUS* REGISTER 11
3976 020164 ROMCLK :NEXT WORD IS INSTRUCTION, BBN
3977 020170 121225 121005!<11*20> :READ FROM IBUS* REGISTER 11
3978 020172 010537 002612 MOV R5,$GDDAT :PUT EXPECTED IN $GDDAT
3979 020176 042737 000222 002612 BIC #222,$GDDAT :CLEAR UNWANTED BITS
3980 020204 116104 000005 MOVB 5(R1),R4 :PUT 'FOUND' INTO R4
3981 020210 042704 000020 BIC #20,R4
3982 020214 123704 002612 CMPB $GDDAT,R4 :DATA CORRECT?
3983 020220 001415 BEQ 65$ :BR IF YES
3984 020222 ERROR 27 :ERROR
3985 020240 104455 TRAP C$ERDF
3986 020242 000033 .WORD 27
3987 020244 005774 .WORD EM27
3988 020246 011230 .WORD ERR27
3989 020250 ESCAPE SEG
3990 020250 104410 TRAP C$ESCAPE
3991 020252 000010 .WORD 10000$-
3992 020254 000241 65$: CLC :CLEAR CARRY
3993 020256 106105 ROLB R5 :SHIFT BIT IN R5
3994 020260 001331 BNE 64$ :IF R5=0 THEN DONE
3995 020262 ENDSEG
3996 020262 10000$:
3997 020262 104405 TRAP C$ESEG
3998 020264 012705 000001 MOV #1,R5 :START WITH BIT 0
3999 020270 BGNSEG
4000 020270 104404 TRAP CSBSEG
4001 020272
4002 020272 005105 67$: COM R5
4003 020274 010561 000004 MOV R5,4(R1) :PUT PATTERN INTO PORT4
4004 020300 042761 000222 000004 BIC #222,4(R1) :CLEAR UNWANTED BITS
4005 020306 ROMCLK :NEXT WORD IS INSTRUCTION, BBN
4006 020312 121111 121100!11 :MOV DATA TO IBUS* REGISTER 11
4007 020314 ROMCLK :NEXT WORD IS INSTRUCTION, BBN
4008 020320 121225 121005!<11*20> :READ FROM IBUS* REGISTER 11
4009 020322 010537 002612 MOV R5,$GDDAT :PUT EXPECTED IN $GDDAT
4010 020326 042737 000222 002612 BIC #222,$GDDAT :CLEAR UNWANTED BITS
```

4011	020334	052737	000020	002612	BIS	#20,\$GDDAT		
4012	020342	116104	000005		MOVB	5(R1),R4		:ADD THESE BITS
4013	020346	123704	002612		CMPB	\$GDDAT,R4		:PUT 'FOUND' INTO R4
4014	020352	001415			BEQ	68\$		:DATA CORRECT?
4015	020354				ERROR	27		:BR IF YES
4016	020372	104455			TRAP	C\$ERDF		:ERROR
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		

```
4033 020422
4034
4035
4036
4037
4038 020422
4039
4040
4041 020422
4042 020422
4043 020422
4044 020426 012737 000000 002576
4045 020434 012705 000001
4046 020440
4047 020444
4048 020444 104404
4049 020446
4050 020446 010561 000004
4051 020452
4052 020456 122100
4053 020460
4054 020464 021005
4055 020466 010537 002612
4056 020472 116104 000005
4057 020476 123704 002612
4058 020502 001413
4059 020504
4060 020522 104455
4061 020524 000035
4062 020526 006054
4063 020530 011450
4064 020532
4065 020532 104410
4066 020534 000010
4067 020536 000241
4068 020540 106105
4069 020542 001341
4070 020544
4071 020544
4072 020544 104405
4073 020546 012705 000001
4074 020552
4075 020552 104404
4076 020554
4077 020554 005105
4078 020556 010561 000004
4079 020562
4080 020566 122100
4081 020570
4082 020574 021005
4083 020576 010537 002612
4084 020602 116104 000005
4085 020606 123704 002612
4086 020612 001413
4087 020614
4088 020632 104455

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

BGNTST
T22::
MSTCLR ;MASTER CLEAR COMM. MICRO-PROCESSOR FAMILY
MOV #0,R0 ;SAVE REGISTER ADDRESS FOR TYPEOUT
MOV #1,R5 ;START WITH BIT 0
MYINT
BGNSEG
TRAP C$BSEG

64$:
MOV R5,4(R1) ;PUT PATTERN INTO PORT4
ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
122100 ;MOV DATA TO IBUS* REGISTER 0
ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
21005 ;READ FROM IBUS* REGISTER 0
MOV R5,$GDDAT ;PUT EXPECTED IN $GDDAT
MOVB 5(R1),R4 ;PUT 'FOUND' INTO R4
CMPB $GDDAT,R4 ;DATA CORRECT?
BEQ 65$ ;BR IF YES
ERROR 29 ;ERROR
TRAP C$ERDF
.WORD 29
.WORD EM29
.WORD ERR29

65$:
ESCAPE SEG
TRAP C$ESCAPE
.WORD 10000$-
CLC ;CLEAR CARRY
ROLB R5 ;SHIFT BIT IN R5
BNE 64$ ;IF R5=0 THEN DONE
ENDSEG

10000$:
TRAP C$ESEG
MOV #1,R5 ;START WITH BIT 0
BGNSEG
TRAP C$BSEG

67$:
COM R5
MOV R5,4(R1) ;PUT PATTERN INTO PORT4
ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
122100 ;MOV DATA TO IBUS* REGISTER 0
ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
21005 ;READ FROM IBUS* REGISTER 0
MOV R5,$GDDAT ;PUT EXPECTED IN $GDDAT
MOVB 5(R1),R4 ;PUT 'FOUND' INTO R4
CMPB $GDDAT,R4 ;DATA CORRECT?
BEQ 68$ ;BR IF YES
ERROR 29 ;ERROR
TRAP C$ERDF
```

4089 020634 000035  
4090 020636 006054  
4091 020640 011450  
4092 020642  
4093 020642 104410  
4094 020644 000012  
4095 020646 005105  
4096 020650 000241  
4097 020652 106105  
4098 020654 001337  
4099 020656  
4100 020656  
4101 020656 104405  
4102 020660  
4103 020660  
4104 020660 104401

68\$: .WORD 29  
.WORD EM29  
.WORD ERR29  
ESCAPE SEG  
TRAP C\$ESCAPE  
.WORD 10001\$-  
COM R5  
CLC  
ROLB R5  
BNE 67\$  
ENDSEG  
10001\$: TRAP C\$ESEG  
ENDTST  
L10076: TRAP C\$ETST

:CHANGE TO FLOATING 1  
:CLEAR CARRY  
:SHIFT BIT IN R5  
:IF R5=0 THEN DONE

```
4105 020662 BADHEAD
4106 :***** TEST 23 *****
4107 :*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
4108 :*FLOAT A 1 THROUGH 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
4116 020666 012737 000001 002576 MSTCLR ;MASTER CLEAR COMM. MICRO-PROCESSOR FAMILY
4117 020674 012705 000001 MOV #1,MRO ;SAVE REGISTER ADDRESS FOR TYPEOUT
4118 020700 MOV #1,R5 ;START WITH BIT 0
4119 020704 MYINT
4120 020704 104404 BGNSEG
4121 020706 64$: TRAP CSBSEG
4122 020706 010561 000004 MOV R5,4(R1) ;PUT PATTERN INTO PORT4
4123 020712 ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
4124 020716 122101 122100!1 ;MOV DATA TO IBUS* REGISTER 1
4125 020720 ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
4126 020724 021025 21005!<1*20> ;READ FROM IBUS* REGISTER 1
4127 020726 010537 002612 MOV R5,$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 CSERDF
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 CSBSEG
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 CSERDF
```

4161 021074 000035  
4162 021076 006054  
4163 021100 011450  
4164 021102  
4165 021102 104410  
4166 021104 000012  
4167 021106 005105  
4168 021110 000241  
4169 021112 106105  
4170 021114 001337  
4171 021116  
4172 021116  
4173 021116 104405  
4174 021120  
4175 021120  
4176 021120 104401

68\$: .WORD 29  
.WORD EM29  
.WORD ERR29  
ESCAPE SEG  
TRAP C\$ESCAPE  
.WORD 10001\$-.  
COM R5 ;CHANGE TO FLOATING 1  
CLC ;CLEAR CARRY  
ROLB R5 ;SHIFT BIT IN R5  
BNE 67\$ ;IF R5=0 THEN DONE  
ENDSEG  
10001\$: TRAP C\$ESEG  
ENDTST  
L10077: TRAP C\$ETST





4233 021334 000035  
4234 021336 006054  
4235 021340 011450  
4236 021342  
4237 021342 104410  
4238 021344 000012  
4239 021346 005105  
4240 021350 000241  
4241 021352 106105  
4242 021354 001337  
4243 021356  
4244 021356  
4245 021356 104405  
4246 021360  
4247 021360  
4248 021360 104401

68\$: .WORD 29  
.WORD EM29  
.WORD ERR29  
ESCAPE SEG  
TRAP C\$ESCAPE  
.WORD 10001\$-  
COM R5  
CLC ;CHANGE TO FLOATING 1  
ROLB R5 ;CLEAR CARRY  
BNE 67\$ ;SHIFT BIT IN R5  
ENDSEG ;IF R5=0 THEN DONE  
10001\$: TRAP C\$ESEG  
ENDTST  
L10100: TRAP C\$ETST

```
4249 021362 BADHEAD
4250 :***** TEST 25 *****
4251 :*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
4252 :*FLOAT A 1 THROUGH 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
4260 021366 012737 002903 002576 MSTCLR :MASTER CLEAR COMM. MICRO-PROCESSOR FAMILY
4261 021374 012705 000601 MOV #3,MRO :SAVE REGISTER ADDRESS FOR TYPEOUT
4262 021400 MYINT #1,R5 :START WITH BIT 0
4263 021404 BGNSEG
4264 021404 104404 TRAP CSBSEG
4265 021406
4266 021406 010561 000004 64$: MOV R5,4(R1) :PUT PATTERN INTO PORT4
4267 021412 ROMCLK :NEXT WORD IS INSTRUCTION, BBN
4268 021416 122103 122100!3 :MOV DATA TO IBUS* REGISTER 3
4269 021420 ROMCLK :NEXT WORD IS INSTRUCTION, BBN
4270 021424 021065 21005!<3*20> :READ FROM IBUS* REGISTER 3
4271 021426 010537 002612 MOV R5,$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 15 :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 CSBSEG
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  
4306 021576 006054  
4307 021600 011450  
4308 021602  
4309 021602 104410  
4310 021604 000012  
4311 021606 005105  
4312 021610 000241  
4313 021612 106105  
4314 021614 001337  
4315 021616  
4316 021616  
4317 021616 104405  
4318 021620  
4319 021620  
4320 021620 104401

68\$: .WORD 29  
.WORD EM29  
.WORD ERR29  
ESCAPE SEG  
TRAP C\$ESCAPE  
.WORD 10001\$-.  
COM R5  
CLC  
ROLB R5  
BNE 67\$  
ENDSEG  
10001\$: TRAP C\$ESEG  
ENDTST  
L10101: TRAP C\$ETST

:CHANGE TO FLOATING 1  
:CLEAR CARRY  
:SHIFT BIT IN R5  
:IF R5=0 THEN DONE

```

4321 021622          BADHEAD
4322                :***** TEST 26 *****
4323                :*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
4324                :*FLOAT A 1 THROUGH 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
4332 021626 012737 000004 002576  MSTCLR          ;MASTER CLEAR COMM. MICRO-PROCESSOR FAMILY
4333 021634 012705 000001          MOV #4,MRO      ;SAVE REGISTER ADDRESS FOR TYPEOUT
4334 021640          MOV #1,R5      ;START WITH BIT 0
4335 021644          MYINT
4336 021644 104404          BGNSEG
4337 021646          TRAP C$BSEG
4338 021646 010561 000004          64$: 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

```

4377 022034 000035  
4378 022036 006054  
4379 022040 011450  
4380 022042 104410  
4381 022042 104410  
4382 022044 000012  
4383 022046 005105  
4384 022050 000241  
4385 022052 106105  
4386 022054 001337  
4387 022056  
4388 022056  
4389 022056 104405  
4390 022060  
4391 022060  
4392 022060 104401

68\$: .WORD 29  
.WORD EM29  
.WORD ERR29  
ESCAPE SEG  
TRAP C\$ESCAPE  
.WORD 10001\$-.  
COM R5  
CLC  
ROLB R5  
BNE 67\$  
ENDSEG  
10001\$: TRAP C\$ESEG  
ENDTST  
L10102: TRAP C\$ETST

:CHANGE TC FLOATING 1  
:CLEAR CARRY  
:SHIFT BIT IN R5  
:IF R5=0 THEN DONE

```
4393 022062
4394
4395
4396
4397
4398 022062
4399
4400
4401 022062
4402 022062
4403 022062
4404 022066 012737 000005 002576
4405 022074 012705 000001
4406 022100
4407 022104
4408 022104 104404
4409 022106
4410 022106 010561 000004
4411 022112
4412 022116 122105
4413 022120
4414 022124 021125
4415 022126 010537 002612
4416 022132 116104 000005
4417 022136 123704 002612
4418 022142 001413
4419 022144
4420 022162 104455
4421 022164 000035
4422 022166 006054
4423 022170 011450
4424 022172
4425 022172 104410
4426 022174 000010
4427 022176 000241
4428 022200 106105
4429 022202 001341
4430 022204
4431 022204
4432 022204 104405
4433 022206 012705 000001
4434 022212
4435 022212 104404
4436 022214
4437 022214 005105
4438 022216 010561 000004
4439 022222
4440 022226 122105
4441 022230
4442 022234 021125
4443 022236 010537 002612
4444 022242 116104 000005
4445 022246 123704 002612
4446 022252 001413
4447 022254
4448 022277 104455

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

BGNTST
T27::
MSTCLR
MOV #5,R0 :MASTER CLEAR COMM. MICRO-PROCESSOR FAMILY
MOV #1,R5 :SAVE REGISTER ADDRESS FOR TYPEOUT
MYINT :START WITH BIT 0
BGNSEG
TRAP CSBSEG

64$:
MOV R5,4(R1) :PUT PATTERN INTO PORT4
ROMCLK :NEXT WORD IS INSTRUCTION, BBN
122100!5 :MOV DATA TO IBUS* REGISTER 5
ROMCLK :NEXT WORD IS INSTRUCTION, BBN
21005!<5*20> :READ FROM IBUS* REGISTER 5
MOV R5,$GDDAT :PUT EXPECTED IN $GDDAT
MOVB 5(R1),R4 :PUT 'FOUND' INTO R4
CMPB $GDDAT,R4 :DATA CORRECT?
BEQ 65$ :BR IF YES
ERROR 29 :ERROR
TRAP CSERDF
.WORD 29
.WORD EM29
.WORD ERR29

65$:
ESCAPE SEG
TRAP C$ESCAPE
.WORD 10000$-.
CLC :CLEAR CARRY
ROLB R5 :SHIFT BIT IN R5
BNE 64$ :IF R5=0 THEN DONE
ENDSEG

10000$:
TRAP C$ESEG
MOV #1,R5 :START WITH BIT 0
BGNSEG
TRAP CSBSEG

67$:
COM R5
MOV R5,4(R1) :PUT PATTERN INTO PORT4
ROMCLK :NEXT WORD IS INSTRUCTION, BBN
122100!5 :MOV DATA TO IBUS* REGISTER 5
ROMCLK :NEXT WORD IS INSTRUCTION, BBN
21005!<5*20> :READ FROM IBUS* REGISTER 5
MOV R5,$GDDAT :PUT EXPECTED IN $GDDAT
MOVB 5(R1),R4 :PUT 'FOUND' INTO R4
CMPB $GDDAT,R4 :DATA CORRECT?
BEQ 68$ :BR IF YES
ERROR 29 :ERROR
TRAP CSERDF
```

4449 022274 000035  
4450 022276 006054  
4451 022300 011450  
4452 022302  
4453 022302 104410  
4454 022304 000012  
4455 022306 005105  
4456 022310 000241  
4457 022312 106105  
4458 022314 001337  
4459 022316  
4460 022316  
4461 022316 104405  
4462 022320  
4463 022320  
4464 022320 104401

68\$: .WORD 29  
.WORD EM29  
.WORD ERR29  
ESCAPE SEG  
TRAP C\$ESCAPE  
.WORD 10001\$-  
COM R5  
CLC :CHANGE TO FLOATING 1  
ROLB R5 :CLEAR CARRY  
BNE 67\$ :SHIFT BIT IN R5  
ENDSEG :IF R5=0 THEN DONE  
10001\$: TRAP C\$ESEG  
ENDTST  
L10103: TRAP C\$ETST



4465	022322				BADHEAD	
4466					:***** TEST 28 *****	
4467					:*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST	
4468					:*FLOAT A 1 THROUGH IBUS REGISTER 6	
4469					:*FLOAT A 0 THROUGH IBUS REGISTER 6	
4470	022322				BADHEAD	
4471					:***** TEST 28 *****	
4472						
4473	022322				BGNTST	
4474	022322				T28::	
4475	022322				MSTCLR	
4476	022326	012737	000006	002576	MOV #6,R0	:MASTER CLEAR COMM. MICRO-PROCESSOR FAMILY
4477	022334	012705	000001		MOV #1,R5	:SAVE REGISTER ADDRESS FOR TYPEOUT
4478	022340				MYINT	:START WITH BIT 0
4479	022344				BGNSEG	
4480	022344	104404			TRAP CSBSEG	
4481	022346					
4482	022346	010561	000004	64\$:	MOV R5,4(R1)	:PUT PATTERN INTO PORT4
4483	022352				ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
4484	022356	122106			122100!6	:MOV DATA TO IBUS* REGISTER 6
4485	022360				ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
4486	022364	021145			21005!<6*20>	:READ FROM IBUS* REGISTER 6
4487	022366	010537	002612		MOV R5,\$GDDAT	:PUT EXPECTED IN \$GDDAT
4488	022372	116104	000005		MOVB 5(R1),R4	:PUT 'FOUND' INTO R4
4489	022376	123704	002612		CMPB \$GDDAT,R4	:DATA CORRECT?
4490	022402	001413			BEQ 65\$	:BR IF YES
4491	022404				ERROR 29	:ERROR
4492	022422	104455			TRAP CSERDF	
4493	022424	000035			.WORD 29	
4494	022426	006054			.WORD EM29	
4495	022430	011450			.WORD ERR29	
4496	022432			65\$:	ESCAPE SEG	
4497	022432	104410			TRAP C\$ESCAPE	
4498	022434	000010			.WORD 10000\$-	
4499	022436	000241			CLC	:CLEAR CARRY
4500	022440	106105			ROLB R5	:SHIFT BIT IN R5
4501	022442	001341			BNE 64\$	:IF R5=0 THEN DONE
4502	022444				ENDSEG	
4503	022444			10000\$:		
4504	022444	104405			TRAP C\$ESEG	
4505	022446	012705	000001		MOV #1,R5	:START WITH BIT 0
4506	022452				BGNSEG	
4507	022452	104404			TRAP CSBSEG	
4508	022454			67\$:		
4509	022454	005105			COM R5	
4510	022456	010561	000004		MOV R5,4(R1)	:PUT PATTERN INTO PORT4
4511	022462				ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
4512	022466	122106			122100!6	:MOV DATA TO IBUS* REGISTER 6
4513	022470				ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
4514	022474	021145			21005!<6*20>	:READ FROM IBUS* REGISTER 6
4515	022476	010537	002612		MOV R5,\$GDDAT	:PUT EXPECTED IN \$GDDAT
4516	022502	116104	000005		MOVB 5(R1),R4	:PUT 'FOUND' INTO R4
4517	022506	123704	002612		CMPB \$GDDAT,R4	:DATA CORRECT?
4518	022512	001413			BEQ 68\$	:BR IF YES
4519	022514				ERROR 29	:ERROR
4520	022532	104455			TRAP CSERDF	

4521 022534 000035  
4522 022536 006054  
4523 022540 011450  
4524 022542  
4525 022542 104410  
4526 022544 000012  
4527 022546 005105  
4528 022550 000241  
4529 022552 106105  
4530 022554 001337  
4531 022556  
4532 022556  
4533 022556 104405  
4534 022560  
4535 022560  
4536 022560 104401

68\$: .WORD 29  
.WORD EM29  
.WORD ERR29  
ESCAPE SEG  
TRAP C\$ESCAPE  
.WORD 10001\$-  
COM R5  
CLC  
ROLB R5  
BNE 67\$  
ENDSEG  
10001\$: TRAP C\$ESEG  
ENDTST  
L10104: TRAP C\$ETST

:CHANGE TO FLOATING 1  
:CLEAR CARRY  
:SHIFT BIT IN R5  
:IF R5=0 THEN DONE

```
4537 022562
4538
4539
4540
4541
4542 022562
4543
4544
4545 022562
4546 022562
4547 022562
4548 022566 012737 000007 002576
4549 022574 012705 000001
4550 022600
4551 022604
4552 022604 104404
4553 022606
4554 022606 010561 000004
4555 022612
4556 022616 122107
4557 022620
4558 022624 021165
4559 022626 010537 002612
4560 022632 116104 000005
4561 022636 123704 002612
4562 022642 001413
4563 022644
4564 022662 104455
4565 022664 000035
4566 022666 006054
4567 022670 011450
4568 022672
4569 022672 104410
4570 022674 000010
4571 022676 000241
4572 022700 106105
4573 022702 001341
4574 022704
4575 022704
4576 022704 104405
4577 022706 012705 000001
4578 022712
4579 022712 104404
4580 022714
4581 022714 005105
4582 022716 010561 000004
4583 022722
4584 022726 122107
4585 022730
4586 022734 021165
4587 022736 010537 002612
4588 022742 116104 000005
4589 022746 123704 002612
4590 022752 001413
4591 022754
4592 022772 104455

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

BGNTST
T29::
MSTCLR
MOV #7,MRO ;MASTER CLEAR COMM. MICRO-PROCESSOR FAMILY
MOV #1,R5 ;SAVE REGISTER ADDRESS FOR TYPEOUT
MYINT ;START WITH BIT 0
BGNSEG
TRAP C$BSEG

64$:
MOV R5,4(R1) ;PUT PATTERN INTO PORT4
ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
122100!7 ;MOV DATA TO IBUS* REGISTER 7
ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
21005!<7*20> ;READ FROM IBUS* REGISTER 7
MOV R5,$GDDAT ;PUT EXPECTED IN $GDDAT
MOVB 5(R1),R4 ;PUT 'FOUND' INTO R4
CMPB $GDDAT,R4 ;DATA CORRECT?
BEQ 65$ ;BR IF YES
ERROR 29 ;ERROR
TRAP C$ERDF
.WORD 29
.WORD EM29
.WORD ERR29

65$:
ESCAPE SEG
TRAP C$ESCAPE
.WORD 10000$-.
CLC ;CLEAR CARRY
ROLB R5 ;SHIFT BIT IN R5
BNE 64$ ;IF R5=0 THEN DONE
ENDSEG

10000$:
TRAP C$ESEG
MOV #1,R5 ;START WITH BIT 0
BGNSEG
TRAP C$BSEG

67$:
COM R5
MOV R5,4(R1) ;PUT PATTERN INTO PORT4
ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
122100!7 ;MOV DATA TO IBUS* REGISTER 7
ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
21005!<7*20> ;READ FROM IBUS* REGISTER 7
MOV R5,$GDDAT ;PUT EXPECTED IN $GDDAT
MOVB 5(R1),R4 ;PUT 'FOUND' INTO R4
CMPB $GDDAT,R4 ;DATA CORRECT?
BEQ 68$ ;BR IF YES
ERROR 29 ;ERROR
TRAP C$ERDF
```

4593 022774 000035  
4594 022776 006054  
4595 023000 011450  
4596 023002  
4597 023002 104410  
4598 023004 000012  
4599 023006 005105  
4600 023010 000241  
4601 023012 106105  
4602 023014 001337  
4603 023016  
4604 023016  
4605 023016 104405  
4606 023020  
4607 023020  
4608 023020 104401

68\$: .WORD 29  
.WORD EM29  
.WORD ERR29  
ESCAPE SEG  
TRAP C\$ESCAPE  
TRAP C\$ESCAPE  
.WORD 10001\$-.  
COM R5  
CLC  
ROLB R5  
BNE 67\$  
ENDSEG  
10001\$: TRAP C\$ESEG  
ENDTST  
L10105: TRAP C\$ETST

;CHANGE TO FLOATING 1  
;CLEAR CARRY  
;SHIFT BIT IN R5  
;IF R5=0 THEN DONE

```
4609 023022
4610
4611
4612
4613
4614 023022
4615
4616
4617 023022
4618 023022
4619 023022
4620 023026 012705 000001
4621 023032 005002
4622 023034
4623 023040
4624 023040 104404
4625 023042 010203
4626 023044 010561 000004
4627 023050 042737 000017 023066
4628 023056 050337 023066
4629 023062
4630 023066 122100
4631 023070 006303
4632 023072 006303
4633 023074 006303
4634 023076 006303
4635 023100 042737 000360 023116
4636 023106 050337 023116
4637 023112
4638 023116 021005
4639 023120 010537 002612
4640 023124 116104 000005
4641 023130 123704 002612
4642 023134 001413
4643 023136
4644 023154 104455
4645 023156 000035
4646 023160 006054
4647 023162 011450
4648 023164
4649 023164 104 10
4650 023166 00LJ14
4651 023170 005205
4652 023172 005202
4653 023174 022702 000010
4654 023200 001320
4655 023202
4656 023202
4657 023202 104405
4658 023204 012705 000001
4659 023210 005002
4660 023212
4661 023212 104404
4662 023214 005003
4663 023216 042737 000360 023234
4664 023224 050337 023234
```

BADHEAD  
:\*\*\*\*\* TEST 30 \*\*\*\*\*  
:\*MICRO PROCESSOR IBUS DUAL ADDRESS TEST  
:\*WRITE ALL IBUS REGISTERS WITH INCREMENTING PATTERN  
:\*READ ALL IBUS REGISTERS TO VERIFY CORRECT ADDRESSING  
BADHEAD  
:\*\*\*\*\* TEST 30 \*\*\*\*\*

BGNTST  
T30::

MSTCLR  
MOV #1,R5 ;MASTER CLEAR COMM. MICRO-PROCESSOR FAMILY  
CLR R2 ;START WITH A ONE  
;R2 CONTAINS ADDRESS OF REGISTER  
MYINT  
BGNSEG  
TRAP C\$BSEG  
1\$: MOV R2,R3 ;R3=REGISTER ADDRESS  
MOV R5,4(R1) ;WRITE DATA TO PORT4  
BIC #17,5\$ ;CLEAR ADDRESS FIELD OF INSTRUCTION  
BIS R3,5\$ ;ADD ADDRESS TO INSTRUCTION  
ROMCLK ;NEXT WORD IS INSTRUCTION, BBN  
5\$: 122100 ;MOVE DATA TO IBUS REGISTER  
ASL R3 ;SHIFT ADDRESS  
ASL R3 ;4 TIMES TO GET  
ASL R3 ;IT TO BITS 4-7  
ASL R3 ;OF NEXT INSTRUCTION  
BIC #360,6\$ ;CLEAR ADDRESS FIELD  
BIS R3,6\$ ;ADD ADDRESS TO INSTRUCTION  
ROMCLK ;NEXT WORD IS INSTRUCTION, BBN  
6\$: 21005 ;READ FROM IBUS REGISTER  
MOV R5,\$GDDAT ;PUT 'EXPECTED' IN \$GDDAT  
MOV B 5(R1),R4 ;PUT 'FOUND' IN R4  
CMPB \$GDDAT,R4 ;IS DATA CORRECT?  
BEQ 2\$ ;BR IF YES  
ERROR 29 ;DATA ERROR  
TRAP C\$ERDF  
.WORD 29  
.WORD EM29  
.WORD ERR29  
2\$: ESCAPE SEG  
TRAP C\$ESCAPE  
.WORD 10000\$-  
INC R5 ;INCREMENT PATTERN  
INC R2 ;INCREMENT REGISTER ADDRESS  
CMP #7+1,R2 ;LAST ADDRESS DONE?  
BNE 1\$ ;BR IF NO  
ENDSEG  
10000\$: TRAP C\$ESEG  
MOV #1,R5 ;RESTART PATTERN TO 1  
CLR R2 ;RESTART AT ADDRESS 0  
BGNSEG  
TRAP C\$BSEG  
3\$: CLR R3 ;RESTART AT ADDRESS 0  
BIC #360,7\$ ;CLEAR ADDRESS FIELD OF INSTRUCTION  
BIS R3,7\$ ;ADD ADDRESS TO INSTRUCTION

4665	023230			ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
4666	023234	021005		21005		:READ FROM IBUS REGISTER
4667	023236	010537	002612	MOV	R5,\$GDDAT	:PUT 'EXPECTED' IN \$GDDAT
4668	023242	116104	000005	MOVB	5(R1),R4	:PUT 'FOUND' IN \$GDDAT
4669	023246	123704	002612	CMPS	\$GDDAT,R4	:DATA CORRECT?
4670	023252	001413		BEQ	4\$	:BR IF YES
4671	023254			ERROR	30	:DUAL ADDRESSING ERROR
4672	023272	104455		TRAP	C\$ERDF	
4673	023274	000036		.WORD	30	
4674	023276	004622		.WORD	EM3C	
4675	023300	011562		.WORD	ERR30	
4676	023302			4\$:	ESCAPE	SEG
4677	023302	104410		TRAP	C\$ESCAPE	
4678	023304	000020		.WORD	10001\$-	
4679	023306	005205		INC	R5	:INCREMENT PATTERN
4680	023310	005202		INC	R2	:NEXT ADDRESS
4681	023312	062703	000020	ADD	#20,R3	:ADD 1 TO ADDRESS IN R3(SHIFTED 4 TIMES)
4682	023316	022702	000010	CMP	#7+1,R2	:LAST ADDRESS DONE?
4683	023322	001335		BNE	3\$	:BR IF NO
4684	023324			ENDSEG		
4685	023324			10001\$:		
4686	023324	104405		TRAP	C\$ESEG	
4687	023326			ENDTST		
4688	023326			L10106:		
4689	023326	104401		TRAP	C\$ETST	



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,MRO		: 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		



4768	023726				BADHEAD	
4769					:***** TEST 32 *****	
4770					:*TEST THE DELAY ON PASS 1 AND REPORT ITS VALUE	
4771	023726				BADHEAD	
4772					:***** TEST 32 *****	
4773						
4774	023726				BGNTST	
4775	023726				T32::	
4776	023726				MSTCLR	: CLEAR M8206
4777	023732				MYINT	
4778	023736	005037	002630		CLR SCLK	: CLEAR FLAG
4779	023742	012761	000020	000004	MOV #20,4(R1)	: HIT TIMER
4780	023750				ROMCLK	
4781	023754	121111			121111	
4782	023756				DELAY 1	: WAIT 100 USEC
4783	023756	012727	000001		MOV #1,(PC)+	
4784	023762	000000			.WORD 0	
4785	023764	013727	002116		MOV LSDLY,(PC)+	
4786	023770	000000			.WORD 0	
4787	023772	005367	177772		DEC -6(PC)	
4788	023776	001375			BNE -4	
4789	024000	005367	177756		DEC -22(PC)	
4790	024004	001367			BNE -20	
4791	024006				ROMCLK	
4792	024012	121225			121005!<11*20>	: READ IBUS* 11
4793	024014	132761	000020	000005	BITB #20,5(R1)	: CHECK CLOCK
4794	024022	001140			BNE 25	: SKIP IF SET
4795	024024				DELAY 250	: WAIT 25 MSEC
4796	024024	012727	000372		MOV #250.,(PC)+	
4797	024030	000000			.WORD 0	
4798	024032	013727	002116		MOV LSDLY,(PC)+	
4799	024036	000000			.WORD 0	
4800	024040	005367	177772		DEC -6(PC)	
4801	024044	001375			BNE -4	
4802	024046	005367	177756		DEC -22(PC)	
4803	024052	001367			BNE -20	
4804	024054				DELAY 250	: WAIT 25 MSEC
4805	024054	012727	000372		MOV #250.,(PC)+	
4806	024060	000000			.WORD 0	
4807	024062	013727	002116		MOV LSDLY,(PC)+	
4808	024066	000000			.WORD 0	
4809	024070	005367	177772		DEC -6(PC)	
4810	024074	001375			BNE -4	
4811	024076	005367	177756		DEC -22(PC)	
4812	024102	001367			BNE -20	
4813	024104				DELAY 250	: WAIT 25 MSEC
4814	024104	012727	000372		MOV #250.,(PC)+	
4815	024110	000000			.WORD 0	
4816	024112	013727	002116		MOV LSDLY,(PC)+	
4817	024116	000000			.WORD 0	
4818	024120	005367	177772		DEC -6(PC)	
4819	024124	001375			BNE -4	
4820	024126	005367	177756		DEC -22(PC)	
4821	024132	001367			BNE -20	
4822	024134				DELAY 250	: WAIT 25 MSEC
4823	024134	012727	000372		MOV #250.,(PC)+	

4824	024140	000000			.WORD	0	
4825	024142	013727	002116		MOV	LSDLY,(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	LSDLY,(PC)+	
4835	024176	000000			.WORD	0	
4836	024200	005367	17.772		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	LSDLY,(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	

CZKMBAD KMC11-B STATIC PART1  
CZKMBAD.P11 20-OCT-81 16:58

MACY11 30A(1052) 21-OCT-81 10:50 L<sup>9</sup> PAGE 115  
HARDWARE TESTS

SEQ 0115

4880	024404	062706	000004
4881	024410		
4882	024410		
4883	024410	104401	

4S:	ADD	#4,SP
L10110:	ENDTST	
	TRAP	CSETST

```
4884 024412
4885
4886
4887
4888
4889 024412
4890
4891
4892 024412
4893 024412
4894
4895 024412
4896 024416 012705 000001
4897 024422
4898 024426
4899 024426 104404
4900 024430
4901 024430 010561 000004
4902 024434
4903 024440 120500
4904 024442
4905 024446 061225
4906 024450 010537 002612
4907 024454 116104 000005
4908 024460 123704 002612
4909 024464 001413
4910 024466
4911 024504 104455
4912 024506 000003
4913 024510 004665
4914 024512 006624
4915 024514
4916 024514 104410
4917 024516 000010
4918 024520 000241
4919 024522 106105
4920 024524 001341
4921 024526
4922 024526
4923 024526 104405
4924 024530 012705 000001
4925 024534
4926 024534
4927 024534 104404
4928 024536
4929 024536 005105
4930 024540 010561 000004
4931 024544
4932 024550 120500
4933 024552
4934 024556 061225
4935 024560 010537 002612
4936 024564 116104 000005
4937 024570 123704 002612
4938 024574 001413
4939 024576

BADHEAD
:***** TEST 33 *****
:*MICRO PROCESSOR BR REGISTER TEST
:*FLOAT A 1 THOUGH THE BR
:*FLOAT A 0 THOUGH THE BR
BADHEAD
:***** TEST 33 *****

BGNTST
T33::

MSTCLR
MOV #1,R5
MYINT
BGNSEG
TRAP C$BSEG

64$:
MOV R5,4(R1)
ROMCLK 120500
ROMCLK 061225
MOV R5,$GDDAT
MOVB 5(R1),R4
CMPB $GDDAT,R4
BEQ 65$
ERROR 3
TRAP C$ERDF
.WORD 3
.WORD EM3
.WORD ERR3

65$:
ESCAPE SEG
TRAP C$ESCAPE
.WORD 10000$-
CLC
ROLB R5
BNE 64$
ENDSEG

10000$:
TRAP C$ESEG
MOV #1,R5

69$:
BGNSEG
TRAP C$BSEG

67$:
COM R5
MOV R5,4(R1)
ROMCLK 120500
ROMCLK 061225
MOV R5,$GDDAT
MOVB 5(R1),R4
CMPB $GDDAT,R4
BEQ 68$
ERROR 3

:R1 CONTAINS BASE COMM. MICRO-PROCESSOR FAMILY ADDRESS
:MASTER CLEAR COMM. MICRO-PROCESSOR FAMILY
:START PATTERN WITH BIT0

:WRITE PATTERN IN PORT4
:NEXT WORD IS INSTRUCTION, BBN
:MOVE DATA TO THE BR REGISTER
:NEXT WORD IS INSTRUCTION, BBN
:MOVE BR TO PORT 5
:PUT 'EXPECTED' IN $GDDAT
:PUT 'FOUND' IN R4
:IS DATA CORRECT?
:BR IF YES
:DATA ERROR

:START PATTERN WITH BIT0

:WRITE PATTERN IN PORT4
:NEXT WORD IS INSTRUCTION, BBN
:MOVE DATA TO THE BR REGISTER
:NEXT WORD IS INSTRUCTION, BBN
:MOVE BR TO PORT 5
:PUT 'EXPECTED' IN $GDDAT
:PUT 'FOUND' IN $GDDAT
:DATA CORRECT?
:BR IF YES
: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								:FAILED TO CLEAR
4962	024706			70\$:				:BRG
4963	024706	005105			COM	R5		:CHANGE BACK TO A ONE
4964	024710	000241			CLC			:CLEAR CARRY
4965	024712	106105			ROLB	R5		:SHIFT BIT IN R5
4966	024714	001310			BNE	67\$		:DONE IF R5=0
4967	024716				ENDSEG			
4968	024716			10001\$:				
4969	024716	104405			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-PROCESSOR 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\$SUB		
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





CZKMBAO KMC11-B STATIC PART1  
CZKMB.A.P11 20-OCT-81 16:58

MACY11 30A(1052) 21-OCT-81 10:50 <sup>D 10</sup> PAGE 120  
HARDWARE TESTS

SEQ 0120

5091 025336 050237 025346  
5092 025342

BIS R2,3\$  
ROMCLK

;ADD ADDRESS TO INSTRUCTION  
;NEXT WORD IS INSTRUCTION, BBN





```
5149 025570
5150
5151
5152
5153 025570
5154
5155
5156 025570
5157 025570
5158 025570
5159 025574 004737 004146
5160 025600 000005
5161 025602 005011
5162 025604 004537 003604
5163 025610 025730
5164 025612 025702
5165 025614 000340 000340
5166 025620
5167 025620 012700 000340
5168 025624 104441
5169 025626 012761 000200 000004
5170 025634
5171 025640 121111
5172 025642
5173 025642 012700 000000
5174 025646 104441
5175 025650 000240
5176 025652
5177 025670 104455
5178 025672 000037
5179 025674 005561
5180 025676 011670
5181 025700 000415
5182 025702
5183 025720 104455
5184 025722 000040
5185 025724 005610
5186 025726 011746
5187 025730 062706 000004
5188 025734
5189 025734
5190 025734
5191 025734 104401
5192
5193 025736
5194
5195
5196
5197 025736
5198
5199
5200 025736
5201 025736
5202 025736
5203 025742
5204 025746 004537 003604
```

BADHEAD  
:\*\*\*\*\* TEST 36 \*\*\*\*\*  
:\*INTERRUPT TEST  
:\*TEST THAT DEVICE CAN INTERRUPT TO VECTOR A  
BADHEAD  
:\*\*\*\*\* TEST 36 \*\*\*\*\*

BGNTST  
T36::

MYINT  
JSR PC,PRDEL ; WAIT FOR ANY PRINTOUTS TO CLEAR TERMINAL  
RESET ;BUS RESET  
CLR (R1) ;CLEAR RUN  
JSR R5,SETVEC ;SET UP VECTORS  
3\$ ;XX0  
2\$ ;XX4  
.WORD 340,340 ;LEVEL 7  
1\$: SETPRI #PRI07 ;PS = LEVEL 7  
MOV #PRI07,R0  
TRAP C\$SPRI  
MOV #200,4(R1) ;WRITE PORT4  
ROMCLK ;NEXT WORD IS INSTRUCTION, BBN  
121111 ;SET BR R0 IN IBUS\* REG 11  
SETPRI #PRI00 ;ALLOW INTERRUPT  
MOV #PRI00,R0  
TRAP C\$SPRI  
NOP  
ERROR 31 ;NO INTERRUPT  
TRAP C\$ERDF  
.WORD 31  
.WORD EM31  
.WORD ERR31  
BR 4\$  
2\$: ERROR 32 ;WRONG VECTOR  
TRAP C\$ERDF  
.WORD 32  
.WORD EM32  
.WORD ERR32  
3\$: ADD #4,SP ;RESET STACK  
4\$:  
ENDTST  
L10115:  
TRAP C\$ETST

BADHEAD  
:\*\*\*\*\* TEST 37 \*\*\*\*\*  
:\*INTERRUPT TEST  
:\*TEST THAT DEVICE CAN INTERRUPT TO VECTOR B  
BADHEAD  
:\*\*\*\*\* TEST 37 \*\*\*\*\*

BGNTST  
T37::

MYINT  
MSTCLR ;MASTER CLEAR M8200,4,6,7  
JSR R5,SETVEC ;SET UP VECTORS



5261	026154	000340	000340			.WORD	340,340		:PRIORITY 7
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	090041				.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			:MASTER CLEAR M8200,4,6,7
5296	026262	012704	000340			MOV	#340,R4		:PUT LEVEL 7 IN R2
5297	026266					SETPRI	R4		:SET PRIORITY TO 7
5298	026266	010400				MOV	R4,R0		
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			BR	1\$		
5325	026406	012716	026404	6\$:	MOV	#6\$, (SP)		:SET UP FOR RTI
5326	026412	000002		2\$:	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		



5429	02700C	000000			2\$: 0		:OUT BA
5430	027002	000000			3\$: 0		:IN BA
5431	027004				ENDTST		
5432	027004				L10122:		
5433	027004	104401			TRAP	CSETST	
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		
5446	027016	005061	000004		CLR	4(R1)	:MASTER CLEAR M8200,4,6,7
5447	027022	004537	003626		JSR	R5,NPRSET	:CLR PORT4
5448	027026	000000			0		:SET UP IBUS REG 0-7
5449	027030	177777			-1		:IN DATA
5450	027032	027152			3\$		:OUT DATA
5451	027034	027151			2\$+1		:IN BA
5452	027036	005037	027150		CLR	2\$	:OUT BA
5453	027042	005061	000004		CLR	4(R1)	:CLEAR 2\$
5454	027046				ROMCLK		:CLEAR PORT 4
5455	027052	121111			121111		:NOW MOVE IT TO IBUS*<11>
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			BEG	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				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	CSETST	
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
5487
5488
5489 027156
5490 027156
5491 027156
5492 027162
5493 027166 013737 002714 027214
5494 027174 013737 002714 027212
5495 027202 004537 003626
5496 027206 000000
5497 027210 125252
5498 027212 000000
5499 027214 000000
5500 027216 012761 000014 000004
5501 027224
5502 027230 121111
5503 027232 012761 000021 000004
5504 027240 012711 003000
5505 027244 012761 121110 000006
5506 027252 052711 00040C
5507 027256 000240
5508 027260 012737 121110 002612
5509 027266 000240
5510 027270 000240
5511
5512
5513
5514
5515
5516
5517
5518
5519
5520
5521
5522
5523 027272
5524 027276 021044
5525 027300
5526 027304 021065
5527 027306 016104 000004
5528 027312 023704 002612
5529 027316 001413
5530 027320
5531 027336 104455
5532 027340 000013
5533 027342 005123
5534 027344 007514
5535 027346
5536 027346
5537 027346
5538 027346 104401
5539
5540 027350

```

```

;*VERIFY CORRECT RESULTS
BADHEAD
;***** TEST 43 *****

```

```

BGNTST
T43::

```

```

MSTCLR
MYINT
MOV KMP06,1$
MOV KMP06,2$
JSR R5,NPRSET
0
125252
0
0
MOV #14,4(R1)
ROMCLK 121111
MOV #21,4(R1)
MOV #BIT9!BIT10,(R1)
MOV #121110,6(R1)
BIS #BIT8,(R1)
NOP
MOV #121110,$GDDAT
NOP
NOP

```

```

:MASTER CLEAR M8200,4,6,7
:USE SEL4 FOR ADDRESS
:USE SEL4 FOR ADDRESS
:LOAD BA AND DATA
:IN DATA
:OUT DATA
:IN BA
:OUT BA
:LOAD SEL 4 WITH CUT BA16 AND 17
:NEXT WORD IS INSTRUCTION, BBN
:SET OUTBA 16 AND 17
:LOAD SEL4
:PUT INSTRUCTION IN SEL6
:CLOCK IT!
:WAIT FOR NPR
:PUT 'EXPECTED' IN $GDDAT

```

```

:OK,LISTEN UP!EXPLANATION TIME.
:ON THE NPR OUT,THE DATA ENDED UP
:IN THE IBUS(NOT IBUS*) SENCE SEL A
:WAS ONLY SELECTED IN THE NPR CYCLE.
:THAT IS,WE DIDN'T REALLY DO AN NPR TO
:PORT 6,THE NPR OUT REALLY ENDED UP IN
:OUT DATA LOW,AND OUT DATA HIGH
:(IBUS <2> AND IBUS <3>).

```

```

:WHAT WE'RE DOING NEXT IS READING IBUS 2&3
:TO SEE IF THE DATA GOT XFERRERD CORRECTLY.

```

```

:READ IBUS <2> PUT IN PORT 4
:READ IBUS <3> PUT IN PORT 5
:PUT 'FOUND' IN R4
:CORRECT RESULTS?
:BR IF YES
:ERROR BA 16 AND 17 FAILED

```

```

3$:
ENDTST
L10124:

```

```

TRAP C$ETST
BADHEAD

```



5541  
5542  
5543  
5544  
5545  
5546  
5547  
5548  
5549 027350  
5550  
5551  
5552 027350  
5553 027350  
5554 027350  
5555 027354  
5556 027360 012737 027546 000004  
5557 027366 012737 000340 000006  
5558 027374 005737 177560  
5559 027400 012737 177560 027426  
5560 027406 012737 177560 027424  
5561 027414 004537 003626  
5562 027420 000000  
5563 027422 125252  
5564 027424 000000  
5565 027426 000000  
5566 027430 012761 000015 000004  
5567 027436 012711 003000  
5568 027442 012761 121110 000006  
5569 027450 052711 000400  
5570 027454 000240  
5571 027456  
5572 027462 021004  
5573 027464  
5574 027470 021025  
5575 027472 016104 000004  
5576 027476 013737 177560 002612  
5577 027504 042737 000200 002612  
5578 027512 023704 002612  
5579 027516 001415  
5580 027520  
5581 027536 104455  
5582 027540 000013  
5583 027542 005123  
5584 027544 007514  
5585 027546  
5586 027546 062706 000004  
5587 027552 013737 002626 000006  
5588 027560 013737 002624 000004  
5589 027566  
5590 027566  
5591 027566 104401  
5592  
5593 027570  
5594

```
***** TEST 44 *****  
: *TEST OF EA BITS 16 AND 17  
: *DO A DATA USING IN BA BITS 16 AND 17  
: *VERIFY CORRECT RESULTS  
: *IN ORDER TO DO THIS TEST, WE WILL READ THE DATA FROM THE  
: *CONSOL TTY CSR IF ONE EXISTS  
: *IF NO CONSOL TTY CSR AT ADDRESS 177560, THIS TEST  
: *WILL BE SKIPPED  
BADHEAD  
***** TEST 44 *****  
BGNTST  
T44::  
MYINT  
MSTCLR  
STOP: MOV #TOUTT,4 :MASTER CLEAR M8200,4,6,7  
MOV #340,6 ;TTY AT ADDRESS 177560 :SET UP FOR TRAP IN CASE IF NO  
TST 177560 :ADDRESS THE TTY-TRAPS HERE IF NONE.  
MOV #177560,1$ :USE SEL4 FOR ADDRESS  
MOV #177560,2$ :USE SEL4 FOR ADDRESS  
JSR R5,NPRSET :LOAD BA AND DATA  
0 :IN DATA  
125252 :OUT DATA  
2$: 0 :IN BA  
1$: 0 :OUT BA  
MOV #15,4(R1)  
MOV #BIT9!BIT10,(R1) :SET CROMI AND CROMO!!  
MOV #121110,6(R1) :PUT INSTR INTO SEL6 MW*  
BIS #BIT8,(R1) :CLOCK IT!  
NOP :WAIT FOR NPR  
ROMCLK :NEXT WORD IS INSTRUCTION, BBN  
021004 :MOVE OUT DATA LB TO SEL4  
ROMCLK :NEXT WORD IS INSTRUCTION, BBN  
021025 :MOVE OUT DATA HB TO SEL5  
MOV 4(R1),R4 :PUT "FOUND" IN R4  
MOV 177560,$GDDAT  
BIC #200,$GDDAT  
CMP $GDDAT,R4 :CORRECT RESULTS?  
BEQ TOUTP :BR IF YES  
ERROR 11 :ERROR BA 16 AND 17 FAILED  
TRAP C$ERDF  
.WORD 11  
.WORD EM11  
.WORD ERR11  
3$:  
TOUTT: ADD #4,SP :UPDATE STACK POINTNER  
TOUTP: MOV SAVE6,6 :RESTORE TRAP VECTOR  
MOV SAVE4,4  
ENDTST  
L10125:  
TRAP C$ETST  
BADHEAD  
***** TEST 45 *****
```

```
5595  
5596  
5597  
5598 027570  
5599  
5600  
5601 027570  
5602 027570  
5603 027570  
5604 027574  
5605 027600 004537 003626  
5606 027604 000000  
5607 027606 000000  
5608 027610 177320  
5609 027612 177320  
5610 027614 012761 000014 000004
```

BGNTST  
T45::

```
MYINT  
MSTCLR  
JSR R5,NPRSET  
0  
0  
177320  
177320  
MOV #14,4(R1)
```

```
:*NPR NON-EXISTENT MEMORY TEST  
:*DO A DATO TO A NON-EXISTENT ADDRESS  
:*VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11  
BADHEAD  
:***** TEST 45 *****  
:MASTER CLEAR M8200.4,6,7  
:LOAD IBUS REGISTERS 0-7  
:IN DATA  
:OUT DATA  
:IN BA  
:IN BA  
:SET OUT BA BITS 16+17 IN PORT4
```





5723	030262	123704	002612		CMPB	\$GDDAT,R4			
5724	030266	001413			BEQ	3\$			:IS DATA CORRECT?
5725	030270				ERROR	11			:BR IF YES
5726	030306	104755			TRAP	C\$ERDF			:ERROR, DATA INCORRECT
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		6\$:	INC	R2			:INC BA
5740	030344	023702	002556		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



5835	030614					ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5836	030620	010000				010000		:LOAD MAR
5837	030622	042737	000017	030640	2\$:	BIC #17,3\$		:CLEAR ADDRESS OF INSTRUCTION
5838	030630	050537	030640			BIS R5,3\$		:ADD ADDRESS TO INSTRUCTION
5839	030634					ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5840	030640	040620			3\$:	040400!<11*20>		:BR SEL B
5841	030642					ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5842	030646	061224				61224		:MOVE BR TO PORT4
5843	030650	111237	002612			MOVB (R2), \$GDDAT		:PUT 'EXPECTED' IN \$GDDAT
5844	030654	116104	000004			MOVB 4(R1), R4		:PUT 'FOUND' IN R4
5845	030660	123704	002612			CMPB \$GDDAT, R4		:DATA CORRECT?
5846	030664	001413				BEQ 4\$		:BR IF YES
5847	030666					ERROR 15		:ALU ERROR
5848	030704	104455				TRAP C\$ERDF		
5849	030706	000017				.WORD 15		
5850	030710	005223				.WORD EM15		
5851	030712	010114				.WORD ERR15		
5852	030714				4\$:	ESCAPE SEG		
5853	030714	104410				TRAP C\$ESCAPE		
5854	030716	000014				.WORD 10000\$-		
5855	030720	005202				INC R2		:NEXT DATA
5856	030722	005205				INC R5		:NEXT ADDRESS
5857	030724	022705	000010			CMP #10, R5		:DONE YET?
5858	030730	001322				BNE 1\$		:BR IF NO
5859	030732					ENDSEG		
5860	030732				10000\$:			
5861	030732	104405				TRAP C\$ESEG		
5862	030734					EXIT TST		
5863	030734	104432				TRAP C\$EXIT		
5864	030736	000012				.WORD L10132-		
5865	030740	000	377	000	5\$:	.BYTE 0,-1,0,-1,125,252,125,252		
5866	030743	377	125	252				
5867	030746	125	252					
5868								
5869								
5870	030750				.EVEN			
5871	030750				ENDTST			
5872	030750	104401			L10132:	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			JSR	PC, CLRC	: CLEAR C BIT!
5897	031012	042737	000017	031030	1\$:	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				010000		: LOAD MAR
5901	031032	042737	000017	031050	2\$:	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			MOVW	(R2), \$GDDAT	: PUT 'EXPECTED' IN \$GDDAT
5908	031064	116104	000004			MOVW	4(R1), R4	: PUT 'FOUND' IN R4
5909	031070	123704	002612			CMPE	\$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 *****		





```
6003 :***** TEST 52 *****
6004 :*ALU TEST
6005 :*TEST OF ALU FUNCTION A AND B WITH C BIT CLEARED
6006 :*ALU FUNCTION (A AND B) CODE=13
6007 :*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6008 :*PERFORM THE FUNCTION, VERIFY THE RESULTS
6009 031372 BADHEAD
6010 :***** TEST 52 *****
6011
6012 031372 BGNTST
6013 031372 T52::
6014 031372
6015 031376 MYINT
6016 031402 005005 MSTCLR ;MASTER CLEAR M8200,4,6,7
6017 031404 012702 031570 CLR R5 ;MEM + SP ADDRESS
6018 031410 004737 003672 MOV #5$,R2 ;POINTER TO CORRECT DATA
6019 031414 002642 JSR PC,MEMLD ;LOAD 8 WORDS OF MAIN MEMORY
6020 031416 004737 004044 MEMDAT ;POINTER TO DATA
6021 031422 002652 JSR PC,SPLD ;LOAD 8 WORDS OF SP
6022 031424 SPDAT ;POINTER TO DATA
6023 031424 104404 BGNSEG
6024 031426 004737 004112 TRAP C$BSEG
6025 031432 042737 000017 031450 1$: JSR PC,CLRC ;CLEAR C BIT!
6026 031440 050537 031450 BIC #17,2$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
6027 031444 ROMCLK ;ADD ADDRESS TO INSTRUCTION
6028 031450 010000 010000 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6029 031452 042737 000017 031470 2$: BIC #17,3$ ;LOAD MAR
6030 031460 050537 031470 BIS R5,3$ ;CLEAR ADDRESS OF INSTRUCTION
6031 031464 ROMCLK ;ADD ADDRESS TO INSTRUCTION
6032 031470 040660 040400! <13*20> ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6033 031472 ROMCLK ;BR A AND B
6034 031476 061224 61224 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6035 031500 111237 002612 MOVB (R2), $GDDAT ;MOVE BR TO PORT4
6036 031504 116104 000004 MOVB 4(R1), R4 ;PUT 'EXPECTED' IN $GDDAT
6037 031510 123704 002612 CMPB $GDDAT, R4 ;PUT 'FOUND' IN R4
6038 031514 001413 BEQ 4$ ;DATA CORRECT?
6039 031516 ERROR 15 ;BR IF YES
6040 031534 104455 TRAP C$ERDF ;ALU ERROR
6041 031536 000017 .WORD 15
6042 031540 005223 .WORD EM15
6043 031542 010114 .WORD ERR15
6044 031544 4$: ESCAPE SEG
6045 031544 104410 TRAP C$ESCAPE
6046 031546 000014 .WORD 10000$-
6047 031550 005202 INC R2 ;NEXT DATA
6048 031552 005205 INC R5 ;NEXT DATA
6049 031554 022705 000010 CMP #10, R5 ;DONE YET?
6050 031560 001322 BNE 1$ ;BR IF NO
6051 031562 ENDSEG
6052 031562 10000$:
6053 031562 104405 TRAP C$ESEG
6054 031564 EXIT TST
6055 031564 104432 TRAP C$EXIT
6056 031566 000012 .WORD L10135-
6057 031570 000 000 000 5$: .BYTE 0,0,0,-1,125,0,0,252
6058 031573 377 125 000
```

6059 031576 00C 252  
6060  
6061  
6062 031600  
6063 031600  
6064 031600 104401  
6065  
6066 031602  
6067  
6068  
6069  
6070  
6071  
6072  
6073 031602  
6074  
6075  
6076 031602  
6077 031602  
6078 031602  
6079 031606  
6080 031612 005005  
6081 031614 012702 032000  
6082 031620 004737 003672  
6083 031624 002642  
6084 031626 004737 004044  
6085 031632 002652  
6086 031634  
6087 031634 104404  
6088 031636 004737 004112  
6089 031642 042737 000017 031660  
6090 031650 050537 031660  
6091 031654  
6092 031660 010000  
6093 031662 042737 000017 031700  
6094 031670 050537 031700  
6095 031674  
6096 031700 040700  
6097 031702  
6098 031706 061224  
6099 031710 111237 002612  
6100 031714 116104 000004  
6101 031720 123704 002612  
6102 031724 001413  
6103 031726  
6104 031744 104455  
6105 031746 006017  
6106 031750 005223  
6107 031752 010114  
6108 031754  
6109 031754 104410  
6110 031756 000014  
6111 031760 005202  
6112 031762 005205  
6113 031764 022705 000010  
6114 031770 001322

.EVEN  
ENDTST  
L10135:

TRAP CSETST

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

BADHEAD

:\*\*\*\*\* TEST 53 \*\*\*\*\*

BGNTST  
T53::

MYINT

MSTCLR

CLR R5

:MASTER CLEAR M8200,4,6,7

MOV #5\$,R2

:MEM + SP ADDRESS

JSR PC,MEMLD

:POINTER TO CORRECT DATA

MEMDAT

:LOAD 8 WORDS OF MAIN MEMORY

JSR PC,SPLD

:POINTER TO DATA

SPDAT

:LOAD 8 WORDS OF SP

BGNSEG

:POINTER TO DATA

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

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! <14\*20>

:BR A OR B

ROMCLK

:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304

61224

:MOVE BR TO PORT4

MOVB (R2),SGDDAT

:PUT 'EXPECTED' IN SGDDAT

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 DATA

CMP #10,R5

:DONE YET?

BNE 1\$

:BR IF NO

6115	031772						ENDSEG	
6116	031772					10000\$:		
6117	031772	104405					TRAP	CSESEG
6118	031774						EXIT	TST
6119	031774	104432					TRAP	CSEXIT
6120	031776	000012					.WORD	L10136-
6121	032000	000	377	377	5\$:		.BYTE	0,-1,-1,-1,125,-1,-1,252
6122	032003	377						
6123	032006	377	252					
6124								
6125								
6126	032010					.EVEN		
6127	032010					ENDTST		
6128	032010	104401				L10136:		
6129							TRAP	CSETST
6130	032012							
6131						BADHEAD		
6132						:*****		
6133						:*ALU TEST		
6134						:*TEST OF ALU FUNCTION A XOR B WITH C BIT CLEARED		
6135						:*ALU FUNCTION (A XOR B) CODE=15		
6136						:*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA		
6137	032012					:*PERFORM THE FUNCTION. VERIFY THE RESULTS		
6138						BADHEAD		
6139						:*****		
6140	032012					BGNTST		
6141	032012					T54::		
6142	032012							
6143	032016					MYINT		
6144	032022	005005				MSTCLR		
6145	032024	012702	032210			CLR	R5	:MASTER CLEAR M8200,4,6,7
6146	032030	004737	003672			MOV	#5\$,R2	:MEM + SP ADDRESS
6147	032034	002642				JSR	PC,MEMLD	:POINTER TO CORRECT DATA
6148	032036	004737	004044			MEMDAT		:LOAD 8 WORDS OF MAIN MEMORY
6149	032042	002652				JSR	PC,SPLD	:POINTER TO DATA
6150	032044					SPDAT		:LOAD 8 WORDS OF SP
6151	032044	104404				BGNSEG		:POINTER TO DATA
6152	032046	004737	004112			TRAP	C\$BSEG	
6153	032052	042737	000017	032070	1\$:	JSR	PC,CLRC	:CLEAR C BIT!
6154	03206C	050537	032070			BIC	#17,2\$	:CLEAR ADDRESS FIELD OF INSTRUCTION
6155	032064					BIS	R5,2\$	:ADD ADDRESS TO INSTRUCTION
6156	032070	010000				ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6157	032072	042737	000017	032110	2\$:	010000		:LOAD MAR
6158	032100	050537	032110			BIC	#17,3\$	:CLEAR ADDRESS OF INSTRUCTION
6159	032104					BIS	R5,3\$	:ADD ADDRESS TO INSTRUCTION
6160	032110	040720				ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6161	032112					040400!	<15*20>	:BR A XOR B
6162	032116	061224				ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6163	032120	111237	002612			61224		:MOVE BR TO PORT4
6164	032124	116104	000004			MOVB	(R2), \$GDDAT	:PUT 'EXPECTED' IN \$GDDAT
6165	032130	123704	002612			MOVB	4(R1), R4	:PUT 'FOUND' IN R4
6166	032134	001413				CMPB	\$GDDAT, R4	:DATA CORRECT?
6167	032136					BEQ	4\$	:BR IF YES
6168	032154	104455				ERROR	15	:ALU ERROR
6169	032156	000017				TRAP	C\$ERDF	
6170	032160	005223				.WORD	15	
						.WORD	EM15	





6283	032504					ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6284	032510	010000				010000		:LOAD MAR
6285	032512	042737	000017	032530	2\$:	BIC #17,3\$		:CLEAR ADDRESS OF INSTRUCTION
6286	032520	050537	032530			BIS R5,3\$		:ADD ADDRESS TO INSTRUCTION
6287	032524					ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6288	032530	040540			3\$:	040400!<6*20>		:BR 2A W/C
6289	032532					ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6290	032536	061224				61224		:MOVE BSR TO PORT4
6291	032540	111237	002612			MOVB (R2),SGDDAT		:PUT 'EXPECTED' IN SGDDAT
6292	032544	116104	000004			MOVB 4(R1),R4		:PUT 'FOUND' IN R4
6293	032550	123704	002612			CMPB SGDDAT,R4		:DATA CORRECT?
6294	032554	001413				BEG 4\$		:BR IF YES
6295	032556					ERROR 15		:ALU ERROR
6296	032574	104455				TRAP C\$ERDF		
6297	032576	000017				.WORD 15		
6298	032600	005223				.WORD EM15		
6299	032602	010114				.WORD ERR15		
6300	032604				4\$:	ESCAPE SEG		
6301	032604	104410				TRAP C\$ESCAPE		
6302	032606	000014				.WORD 10000\$-		
6303	032610	005202				INC R2		:NEXT DATA
6304	032612	005205				INC R5		:NEXT ADDRESS
6305	032614	022705	000010			CMP #10,R5		:DONE YET?
6306	032620	001322				BNE 1\$		:BR IF NO
6307	032622					ENDSEG		
6308	032622				10000\$:	TRAP C\$ESEG		
6309	032622	104405				EXIT TST		
6310	032624					TRAP C\$EXIT		
6311	032624	104432				.WORD L10141-		
6312	032626	000012			5\$:	.BYTE 0,0,376,376,252,252,124,124		
6313	032630	000	000	376				
6314	032633	376	252	252				
6315	032636	124	124					
6316								
6317								
6318	032640				.EVEN			
6319	032640				ENDTST			
6320	032640	104401			L10141:	TRAP C\$ETST		
6321								
6322	032642					BADHEAD		
6323						:***** TEST 57 *****		
6324						:*ALU TEST		
6325						:*TEST OF ALU FUNCTION SUB WITH C BIT CLEARED		
6326						:*ALU FUNCTION (A-B) CODE=16		
6327						:*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA		
6328						:*PERFORM THE FUNCTION, VERIFY THE RESULTS		
6329	032642					BADHEAD		
6330						:***** TEST 57 *****		
6331								
6332	032642				BGNTST			
6333	032642				T57::			
6334	032642					MYINT		
6335	032646					MSTCLR		:MASTER CLEAR M8200,4,6,7
6336	032652	005005				CLR R5		:MEM + SP ADDRESS
6337	032654	012702	033042			MOV #5\$,R2		:POINTER TO CORRECT DATA
6338	032660	004737	003672			JSR PC,MEMLD		:LOAD 8 WORDS OF MAIN MEMORY

```
6339 032664 002642 MEMDAT
6340 032666 004737 004044 JSR PC,SPLD ;POINTER TO DATA
6341 032672 002652 SPDAT ;LOAD 8 WORDS OF SP
6342 032674 BGNSEG ;POINTER TO DATA
6343 032674 104404 TRAP C$BSEG
6344 032676 004737 004112 JSR PC,CLRC ;CLEAR C BIT!
6345 032702 042737 000017 032720 1$: 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 010000 2$: ;LOAD MAR
6349 032722 042737 000017 032740 BIC #17,3$ ;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
6389
6390 BADHEAD
6391 ;***** TEST 58 *****
6392 ;*ALU TEST
6393 ;*TEST OF ALU FUNCTION ADD W/C WITH C BIT CLEARED
6394 ;*ALU FUNCTION (A PLUS B PLUS C) CODE=01
;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
;*PERFORM THE FUNCTION, VERIFY THE RESULTS
```



```

6395 033054
6396
6397
6398 033054
6399 033054
6400 033054
6401 033060
6402 033064 005005
6403 033066 012702 033252
6404 033072 004737 003672
6405 033076 002642
6406 033100 004737 004044
6407 033104 002652
6408 033106
6409 033106 104404
6410 033110 004737 004112
6411 033114 042737 000017 033132 1$:
6412 033122 050537 033132
6413 033126
6414 033132 010000
6415 033134 042737 000017 033152 2$:
6416 033142 050537 033152
6417 033146
6418 033152 040420
6419 033154
6420 033160 061224
6421 033162 111237 002612
6422 033166 116104 000004
6423 033172 123704 002612
6424 033176 001413
6425 033200
6426 033216 104455
6427 033220 000017
6428 033222 005223
6429 033224 010114
6430 033226
6431 033226 104410
6432 033230 000014
6433 033232 005202
6434 033234 005205
6435 033236 022705 000010
6436 033242 001322
6437 033244
6438 033244
6439 033244 104405
6440 033246
6441 033246 104432
6442 033250 000012
6443 033252 000 377 377 5$:
6444 033255 376 252 377
6445 033260 377 124
6446
6447
6448 033262
6449 033262
6450 033262 104401

BADHEAD
:***** TEST 58 *****

BGNTST
T58::
MYINT
MSTCLR
CLR R5 ;MASTER CLEAR M8200,4,6,7
MOV #5$,R2 ;MEM + SP ADDRESS
JSR PC,MEMLD ;POINTER TO CORRECT DATA
MEMDAT ;LOAD 8 WORDS OF MAIN MEMORY
JSR PC,SPLD ;POINTER TO DATA
SPDAT ;LOAD 8 WORDS OF SP
BGNSEG ;POINTER TO DATA
TRAP C$BSEG
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
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
040400!<01*20> ;BR ADD W/C
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
61224 ;MOVE BR TO PORT4
MOV# (R2), $GDDAT ;PUT 'EXPECTED' IN $GDDAT
MOV# 4(R1), R4 ;PUT 'FOUND' IN R4
CMP# $GDDAT, R4 ;DATA CORRECT?
BEQ 4$ ;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-
.BYTE 0,-1,-1,376,252,-1,-1,124

.EVEN
ENDTST
L10143: TRAP C$ETST

```

6451  
6452  
6453 033264  
6454  
6455  
6456  
6457  
6458  
6459  
6460 033264  
6461  
6462  
6463 033264  
6464 033264  
6465 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  
BADHEAD  
:\*\*\*\*\* TEST 59 \*\*\*\*\*

BGNTST  
T59::  
MYINT



```
6522      ;*TEST OF ALU FUNCTION INC A WITH C BIT CLEARED
6523      ;*ALU FUNCTION (A PLUS 1)          CODE=3
6524      ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
6525      ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
6526 033474 BADHEAD
6527      ;***** TEST 60 *****
6528
6529 033474      BGNTST
6530 033474      T60::
6531 033474
6532 033500      MYINT
6533 033504 012702 033672      MSTCLR      ;MASTER CLEAR M8200,4,6,7
6534 033510 005005      MOV #5$,R2      ;POINTER TO CORRECT DATA
6535 033512 004737 003672      CLR R5
6536 033516 002642      JSR PC,MEMLD      ;LOAD 8 WORDS OF MAIN MEMRY
6537 033520 004737 004044      MEMDAT      ;POINTER TO DATA
6538 033524 002652      JSR PC,SPLD      ;LOAD 8 WORDS OF SP
6539 033526      SPDAT      ;POINTER TO DATA
6540 033526 104404      BGNSEG
6541 033530 004737 004112      TRAP C$BSEG
6542 033534 042737 000017 033552 1$: JSR PC,CLRC      ;CLEAR C BIT!
6543 033542 050537 033552      BIC #17,2$      ;CLEAR ADDRESS FIELD OF INSTRUCTION
6544 033546      BIS R5,2$      ;ADD ADDRESS TO INSTRUCTION
6545 033552 010000      ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6546 033554 042737 000017 033572 2$: 010000      ;LOAD MAR
6547 033562 050537 033572      BIC #17,3$      ;CLEAR ADDRESS OF INSTRUCTION
6548 033566      BIS R5,3$      ;ADD ADDRESS TO INSTRUCTION
6549 033572 040460      ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6550 033574      040400!<3*20>      ;BR INC A
6551 033600 061224      ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
6552 033602 111237 002612      61224      ;MOVE BR TO PORT4
6553 033606 116104 000004      MOVB (R2), $GDDAT      ;PUT 'EXPECTED' IN $GDDAT
6554 033612 123704 002612      MOVB 4(R1), R4      ;PUT 'FOUND' IN R4
6555 033616 001413      CMPB $GDDAT, R4      ;DATA CORRECT?
6556 033620      BEQ 4$      ;BR IF YES
6557 033636 104455      ERROR 15      ;ALU ERROR
6558 033640 000017      TRAP C$ERDF
6559 033642 005223      .WORD 15
6560 033644 010114      .WORD EM15
6561 033646      .WORD ERR15
6562 033646 104410      4$: ESCAPE SEG
6563 033650 000014      TRAP C$ESCAPE
6564 033652 005202      .WORD 10000$-.
6565 033654 005205      INC R2      ;NEXT DATA
6566 033656 022705 000010      INC R5
6567 033662 001322      CMP #10, R5      ;DONE YET?
6568 033664      BNE 1$      ;BR IF NO
6569 033664      10000$: ENDSEG
6570 033664 104405      TRAP C$ESEG
6571 033666      EXIT TST
6572 033666 104432      TRAP C$EXIT
6573 033670 000012      .WORD L10145-.
6574 033672 001 001 000 5$: .BYTE 1,1,0,0,126,126,253,253
6575 033675 000 126 126
6576 033700 253 253
6577
```

```
6578  
6579 033702  
6580 033702  
6581 033702 104401  
6582  
6583  
6584 033704  
6585  
6586  
6587  
6588  
6589  
6590  
6591 033704  
6592  
6593  
6594 033704  
6595 033704  
6596 033704  
6597 033710  
6598 033714 005005  
6599 033716 012702 034102  
6600 033722 004737 003672  
6601 033726 002642  
6602 033730 004737 004044  
6603 033734 002652  
6604 033736  
6605 033736 104404  
6606 033740 004737 004112  
6607 033744 042737 000017 033762 1$:  
6608 033752 050537 033762  
6609 033756  
6610 033762 010000 034002 2$:  
6611 033764 042737 000017  
6612 033772 050537 034002  
6613 033776  
6614 034002 040520 3$:  
6615 034004  
6616 034010 061224  
6617 034012 111237 002612  
6618 034016 116104 000004  
6619 034022 123704 002612  
6620 034026 001413  
6621 034030  
6622 034046 104455  
6623 034050 000017  
6624 034052 005223  
6625 034054 010114  
6626 034056  
6627 034056 104410 4$:  
6628 034060 000014  
6629 034062 005202  
6630 034064 005205  
6631 034066 022705 000010  
6632 034072 001322  
6633 034074
```

```
.EVEN  
ENDTST  
L10145:  
TRAP CSETST  
  
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$,R2 ;POINTER TO CORRECT DATA  
JSR PC,MEMLD ;LOAD 8 WORDS OF MAIN MEMORY  
MEMDAT ;POINTER TO DATA  
JSR PC,SPLD ;LOAD 8 WORDS OF SP  
SPDAT ;POINTER TO DATA  
BGNSEG  
TRAP CSBSEG  
JSR PC,CLRC ;CLEAR C BIT!  
BIC #17,2$ ;CLEAR ADDRESS FIELD OF INSTRUCTION  
BIS R5,2$ ;ADD ADDRESS TO INSTRUCTION  
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304  
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  
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 CSERDF  
.WORD 15  
.WORD EM15  
.WORD ERR15  
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
```

6634	034074				
6635	034074	104405			
6636	034076				
6637	034076	104432			
6638	034100	000012			
6639	034102	000	000	376	5\$:
6640	034105	376	252	252	
6641	034110	124	124		
6642					
6643					
6644	034112				
6645	034112				
6646	034112	104401			
6647					
6648					
6649	034114				
6650					
6651					
6652					
6653					
6654					
6655					
6656	034114				
6657					
6658					
6659	034114				
6660	034114				
6661	034114				
6662	034120				
6663	034124	005005			
6664	034126	012702	034312		
6665	034132	004737	003672		
6666	034136	002642			
6667	034140	004737	004044		
6668	034144	002652			
6669	034146				
6670	034146	104404			
6671	034150	004737	004112		
6672	034154	042737	000017	034172	1\$:
6673	034162	050537	034172		
6674	034166				
6675	034172	010000			2\$:
6676	034174	042737	000017	034212	
6677	034202	050537	034212		
6678	034206				
6679	034212	040500			3\$:
6680	034214				
6681	034220	061224			
6682	034222	111237	002612		
6683	034226	116104	000004		
6684	034232	123704	002612		
6685	034236	001413			
6686	034240				
6687	034256	104455			
6688	034260	000017			
6689	034262	005223			

10000\$:  
 TRAP C\$ESEG  
 EXIT TST  
 TRAP C\$EXIT  
 .WORD L10146-  
 .BYTE 0,0,376,376,252,252,124,124

.EVEN  
 ENDTST  
 L10146:  
 TRAP C\$ETST

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  
 BADHEAD  
 :\*\*\*\*\* TEST 62 \*\*\*\*\*

BGNTST  
 T62::  
 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  
 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  
 010000 :LOAD MAR  
 BIC #17,3\$ :CLEAR ADDRESS OF INSTRUCTION  
 BIS R5,3\$ :ADD ADDRESS TO INSTRUCTION  
 ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=304  
 040400!<4\*20> :BR A PLUS C  
 ROMCLK :NEXT WORD IS INSTRUCTION, ROMCKL PC=5305  
 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



```
6746 034430 061224 61224 :MOVE BR TO PORT4
6747 034432 111237 002612 MOVB (R2),SGDDAT :PUT 'EXPECTED' IN SGDDAT
6748 034436 116104 000004 MOVB 4(R1),R4 :PUT 'FOUND' IN R4
6749 034442 123704 002612 CMPB SGDDAT,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$: TRAP C$ESEG
6765 034514 104405 EXIT TST
6766 034516 104432 TRAP C$EXIT
6767 034516 000012 .WORD L10150-
6768 034520 377 000 376 5$: .BYTE -1,0,376,-1,-1,252,124,-1
6769 034522 377 377 252
6770 034525 124 377
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
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 INSTRUCTION
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			MOVW	(R2),SGDDAT	:PUT 'EXPECTED' IN SGDDAT
6814	034646	116104	000004			MOVW	4(R1),R4	:PUT 'FOUND' IN R4
6815	034652	123704	002612			CMPS	SGDDAT,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								
6840	034742				.EVEN			
6841	034742				ENDTST			
6842	034742	104401			L10151:			
6843					TRAP	C\$ETST		
6844								
6845	034744							
6846					BADHEAD			
6847					:***** TEST 65 *****			
6848					:*ALU TEST			
6849					:*TEST OF ALU FUNCTION SEL B WITH C BIT SET			
6850					:*ALU FUNCTION (B) CODE=11			
6851					:*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA			
6852	034744				:*PERFORM THE FUNCTION, VERIFY THE RESULTS			
6853					BADHEAD			
6854					:***** TEST 65 *****			
6855	034744							
6856	034744				BGNTST			
6857	034744				T65::			
					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	035022	1\$:	JSR	PC,SETC	:SET C BIT!
6868	035004	042737	000017			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:			
6908						TRAP	C\$ETST	
6909								
6910	035154					BADHEAD		
6911						:***** TEST 66 *****		
6912						:*ALU TEST		
6913						:*TEST OF ALU FUNCTION SEL A WITH C BIT SET		

CZKMBAD KMC11-B STATIC PART1  
CZAMBA.P11 20-OCT-81 16:58

MACY11 30A(1052) 21-OCT-81 10:50 M 12  
HARDWARE TESTS PAGE 155

SEQ 0155

6914  
6915  
6916  
6917 035154  
6918

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



```

6975
6976 035364
6977
6978
6979
6980
6981
6982
6983 035364
6984
6985
6986 035364
6987 035364
6988 035364
6989 035370
6990 035374 005005
6991 035376 012702 035562
6992 035402 004737 003672
6993 035406 002642
6994 035410 004737 004044
6995 035414 002652
6996 035416
6997 035416 104404
6998 035420 004737 004130
6999 035424 042737 000017 035442 1$:
7000 035432 050537 035442
7001 035436
7002 035442 010000 035462 2$:
7003 035444 042737 000017 035462
7004 035452 050537 035462
7005 035456
7006 035462 040640 3$:
7007 035464
7008 035470 061224
7009 035472 111237 002612
7010 035476 116104 000004
7011 035502 123704 002612
7012 035506 001413
7013 035510
7014 035526 104455
7015 035530 000017
7016 035532 005223
7017 035534 010114
7018 035536
7019 035536 104410 4$:
7020 035540 000014
7021 035542 005202
7022 035544 005205
7023 035546 022705 000010
7024 035552 001322
7025 035554
7026 035554
7027 035554 104405 10000$:
7028 035556
7029 035556 104432
7030 035560 000012

```

```

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
BADHEAD
:***** TEST 67 *****

```

BGNTST  
T67::

```

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
JSR PC,SETC :SET C BIT!
BIC #17,2$ :CLEAR ADDRESS FIELD OF INSTRUCTION
BIS R5,2$ :ADD ADDRESS TO INSTRUCTION
ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
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
040400!<12*20> :BR A OR NOTB
ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
61224 :MOVE BR TO PORT4
MOVB (R2),SGDDAT :PUT 'EXPECTED' IN SGDDAT
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
10000$: TRAP C$SEEG
EXIT TST
TRAP C$EXIT
.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								
7036	035572							
7037	035572							
7038	035572	104401						
7039								
7040								
7041	035574							
7042								
7043								
7044								
7045								
7046								
7047								
7048	035574							
7049								
7050								
7051	035574							
7052	035574							
7053	035574							
7054	035600							
7055	035604	005005						
7056	035606	012702	035772					
7057	035612	004737	003672					
7058	035616	002642						
7059	035620	004737	004044					
7060	035624	002652						
7061	035626							
7062	035626	104404						
7063	035630	004737	004130					
7064	035634	042737	000017	035652	1\$:			
7065	035642	050537	035652					
7066	035646							
7067	035652	010000						
7068	035654	042737	000017	035672	2\$:			
7069	035662	050537	035672					
7070	035666							
7071	035672	040660			3\$:			
7072	035674							
7073	035700	061224						
7074	035702	111237	002612					
7075	035706	116104	000004					
7076	035712	123704	002612					
7077	035716	001413						
7078	035720							
7079	035736	104455						
7080	035740	000027						
7081	035742	005475						
7082	035744	010624						
7083	035746							
7084	035746	104410						
7085	035750	000014						
7086	035752	005202						

.EVEN  
ENDTST  
L10154:  
TRAP C\$ETST

BADHEAD  
:\*\*\*\*\* TEST 68 \*\*\*\*\*  
:\*ALU TEST  
:\*TEST OF ALU FUNCTION A AND B WITH C BIT SET  
:ALU FUNCTION (A AND B) CODE='3'  
:\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
:\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
BADHEAD  
:\*\*\*\*\* TEST 68 \*\*\*\*\*

BGNTST  
T68::  
MYINT  
MSTCLR  
CLR R5 ;MASTER CLEAR M8200,4,6,7  
MOV #5\$,R2 ;MEM + SP ADDRESS  
JSR PC,MEMLD ;POINTER TO CORRECT ADDRESS  
MEMDAT ;LOAD 8 WORDS OF MAIN MEMORY  
JSR PC,SPLD ;POINTER TO DATA  
SPDAT ;LOAD 8 WORDS OF SP  
BGNSEG ;POINTER TO DATA  
TRAP C\$BSEG  
JSR PC,SETC ;SET C BIT!  
BIC #17,2\$ ;CLEAR ADDRESS FIELD OF INSTRUCTION  
BIS R5,2\$ ;ADD ADDRESS TO INSTRUCTION  
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304  
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  
040400!<13\*20> ;BR A AND B  
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 23 ;ALU ERROR  
TRAP C\$ERDF  
.WORD 23  
.WORD EM23  
.WORD ERR23  
4\$:  
ESCAPE SEG  
TRAP C\$ESCAPE  
.WORD 10000\$-  
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								
7101	036002					.EVEN		
7102	036002					ENDTST		
7103	036002	104401				L10155:		
7104						TRAP	C\$ETST	
7105								
7106	036004					BADHEAD		
7107						:*****		
7108						:***** TEST 69 *****		
7109						:*ALU TEST		
7110						:*TEST OF ALU FUNCTION A OR B WITH C BIT SET		
7111						:*ALU FUNCTION (A OR B) CODE=14		
7112						:*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA		
7113	036004					:*PERFORM THE FUNCTION, VERIFY THE RESULTS		
7114						BADHEAD		
7115						:*****		
7116	036004					:***** TEST 69 *****		
7117	036004							
7118	036004							
7119	036010					BGNTST		
7120	036014	005005				T69::		
7121	036016	012702	036202			MYINT		
7122	036022	004737	003672			MSTCLR		:MASTER CLEAR M8200,4,6,7
7123	036026	002642				CLR	R5	:MEM + SP ADDRESS
7124	036030	004737	004044			MOV	#5\$,R2	:POINTER TO CORRECT DATA
7125	036034	002652				JSR	PC,MEMLD	:LOAD 8 WORDS OF MAIN MEMORY
7126	036036					MEMDAT		:POINTER TO DATA
7127	036036	104404				JSR	PC,SPLD	:LOAD 8 WORDS OF SP
7128	036040	004737	004130			SPDAT		:POINTER TO DATA
7129	036044	042737	000017	036062	1\$:	BGNSEG		
7130	036052	050537	036062			TRAP	C\$BSEG	
7131	036056					JSR	PC,SETC	:SET C BIT!
7132	036062	010000				BIC #17,2\$		:CLEAR ADDRESS FIELD OF INSTRUCTION
7133	036064	042737	000017	036102	2\$:	BIS	R5,2\$	:ADD ADDRESS TO INSTRUCTION
7134	036072	050537	036102			ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7135	036076					010000		:LOAD MAR
7136	036102	040700				BIC #17,3\$		:CLEAR ADDRESS OF INSTRUCTION
7137	036104					BIS	R5,3\$	:ADD ADDRESS TO INSTRUCTION
7138	036110	061224				ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7139	036112	111237	002612			040400!<14*20>		:BR A OR B
7140	036116	116104	000004			ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7141	036122	123704	002612			61224		:MOVE BR TO PORT4
7142	036126	001413				MOVB (R2), \$GDDAT		:PUT 'EXPECTED' IN R4
						MOVB 4(R1), R4		:PUT 'FOUND' IN R4
						CMPB \$GDDAT, R4		:DATA CORRECT?
						BEQ 4\$		:BR IF YES

7143	036130						ERROR	23		:ALU ERROR
7144	036146	104455					TRAP	C\$ERDF		
7145	036150	000027					.WORD	23		
7146	036152	005475					.WORD	EM23		
7147	036154	010624					.WORD	ERR23		
7148	036156						ESCAPE	SEG		
7149	036156	104410				4\$:	TRAP	C\$ESCAPE		
7150	036160	000014					.WORD	10000\$-		
7151	036162	005202					INC	R2		:NEXT DATA
7152	036164	005205					INC	R5		:NEXT ADDRESS
7153	036166	022705	000010				CMP	#10,R5		:DONE YET?
7154	036172	001322					BNE	1\$		:BR IF NO
7155	036174						ENDSEG			
7156	036174					10000\$:				
7157	036174	104405					TRAP	C\$ESEG		
7158	036176						EXIT	TST		
7159	036176	104432					TRAP	C\$EXIT		
7160	036200	000012					.WORD	L10156-		
7161	036202	000	377	377	5\$:	.BYTE	0,-1,-1,-1,125,-1,-1,252			
7162	036205	377		125						
7163	036210	377		252						
7164										
7165							.EVEN			
7166	036212					ENDTST				
7167	036212					L10156:				
7168	036212	104401					TRAP	C\$ETST		
7169										
7170										
7171	036214						BADHEAD			
7172							:***** TEST 70 *****			
7173							:*ALU TEST			
7174							:*TEST OF ALU FUNCTION A XOR B WITH C BIT SET			
7175							:*ALU FUNCTION (A XOR B) CODE=15			
7176							:*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA			
7177							:*PERFORM THE FUNCTION, VERIFY THE RESULTS			
7178										
7179	036214						BADHEAD			
7180							:***** TEST 70 *****			
7181	036214					BGNTST				
7182	036214					T70::				
7183	036214						MYINT			
7184	036220						MSTCLR			
7185	036224	005005					CLR	R5		:MASTER CLEAR M8200,4,6,7
7186	036226	012702	036412				MOV	#5\$,R2		:MEM + SP ADDRESS
7187	036232	004737	003672				JSR	PC,MEMLD		:POINTER TO CORRECT DATA
7188	036236	002642					MEMDAT			:LOAD 8 WORDS OF MAIN MEMORY
7189	036240	004737	004044				JSR	PC,SPLD		:POINTER TO DATA
7190	036244	002652					SPDAT			:LOAD 8 WORDS OF SP
7191	036246						BGNSEG			:POINTER TO DATA
7192	036246	104404					TRAP	C\$BSEG		
7193	036250	004737	004130			1\$:	JSR	PC,SETC		:SET C BIT!
7194	036254	042737	000017	036272			BIC	#17,2\$		:CLEAR ADDRESS FIELD OF INSTRUCTION
7195	036262	050537	036272				BIS	R5,2\$		:ADD ADDRESS TO INSTRUCTION
7196	036266						ROMCLK			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7197	036272	010000				2\$:	010000			:LOAD MAR
7198	036274	042737	000017	036312			BIC	#17,3\$		:CLEAR ADDRESS OF INSTRUCTION





```

7255 036454 002652          SPDAT          :OOINTER TO DATA
7256 036456                BGNSEG
7257 036456 104404          TRAP          C$BSEG
7258 036460 004737 004130    JSR           PC,SETC      :SET C BIT!
7259 036464 042737 000017    BIC           #17,2$      :CLEAR ADDRESS FIELD OF INSTRUCTION
7260 036472 050537 036502    BIS           R5,2$       :ADD ADDRESS TO INSTRUCTION
7261 036476                ROMCLK
7262 036502 010000          010000        :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7263 036504 042737 000017    BIC           #17,3$      :LOAD MAR
7264 036512 050537 036522    BIS           R5,3$       :CLEAR ADDRESS OF INSTRUCTION
7265 036516                ROMCLK
7266 036522 040400          040400!<00*20>        :ADD ADDRESS TO INSTRUCTION
7267 036524                ROMCLK
7268 036530 061224          61224         :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7269 036532 111237 002612    MOVB          (R2),SGDDAT  :BR ADD
7270 036536 116104 000004    MOVB          4(R1),R4     :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7271 036542 123704 002612    CMPB          $GDDAT,R4    :MOVE BR TO PORT4
7272 036546 001413          BEQ           4$          :PUT 'EXPECTED' IN $GDDAT
7273 036550                ERROR          23          :PUT 'FOUND' IN R4
7274 036566 104455          TRAP          C$ERDF      :DATA CORRECT?
7275 036570 000027          .WORD        23          :BR IF YES
7276 036572 005475          .WORD        EM23        :ALU ERROR
7277 036574 010624          .WORD        ERR23
7278 036576                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
7296 036632          .EVEN
7297 036632          ENDTST
7298 036632 104401          L10160:
7299                TRAP          C$ETST
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

```



```

7367
7368
7369
7370
7371
7372
7373 037044
7374
7375
7376 037044
7377 037044
7378 037044
7379 037050
7380 037054 005005
7381 037056 012702 037242
7382 037062 004737 003672
7383 037066 002642
7384 037070 004737 004044
7385 037074 002652
7386 037076
7387 037076 104404
7388 037100 004737 004130
7389 037104 042737 000017 037122 1$:
7390 037112 050537 037122
7391 037116
7392 037122 010000
7393 037124 042737 000017 037142 2$:
7394 037132 050537 037142
7395 037136
7396 037142 040740 3$:
7397 037144
7398 037150 061224
7399 037152 111237 002612
7400 037156 116104 000004
7401 037162 123704 002612
7402 037166 001413
7403 037170
7404 037206 104455
7405 037210 000027
7406 037212 005475
7407 037214 010624
7408 037216
7409 037216 104410 4$:
7410 037220 000014
7411 037222 005202
7412 037224 005205
7413 037226 022705 000010
7414 037232 001322
7415 037234
7416 037234
7417 037234 104405 10000$:
7418 037236
7419 037236 104432
7420 037240 000012
7421 037242 000 001 377 5$:
7422 037245 000 000 253

```

```

:***** 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
BADHEAD
:***** TEST 73 *****

BGNTST
T73::
MYINT
MSTCLR
CLR R5 :MASTER CLEAR M8200,4,6,7
MOV #5$,R2 :MEM + SP ADDRESS
JSR PC,MEMLD :POINTER TO CORRECT DATA
MEMDAT :LOAD 8 WORDS OF MAIN MEMORY
JSR PC,SPLD :POINTER TO DATA
SPDAT :LOAD 8 WORDS OF SP
BGNSEG :POINTER TO DATA
TRAP C$BSEG
JSR PC,SETC :SET C BIT!
BIC #17,2$ :CLEAR ADDRESS FIELD OF INSTRUCTION
BIS R5,2$ :ADD ADDRESS TO INSTRUCTION
ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
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
040400!<16*20> :BR SUB
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 23 :ALU ERROR
TRAP C$ERDF
.WORD 23
.WORD EM23
.WORD ERR23
4$: ESCAPE SEG
TRAP C$ESCAPE
.WORD 10000$-
INC R2 :NEXT DATA
INC R5 :NEXT ADDRESS
CMP #10, R5 :DONE YET?
BNE 1$ :BR IF NO
ENDSEG
10000$: TRAP C$ESEG
EXIT TST
TRAP C$EXIT
.WORD L10162-
.BYTE 0,1,-1,0,0,253,125,0

```

7423	037250	125	000						
7424									
7425									
7426	037252								
7427	037252								
7428	037252	104401							
7429									
7430									
7431	037254								
7432									
7433									
7434									
7435									
7436									
7437									
7438	037254								
7439									
7440									
7441	037254								
7442	037254								
7443	037254								
7444	037260								
7445	037264	005005							
7446	037266	012702	037452						
7447	037272	004737	003672						
7448	037276	002642							
7449	037300	004737	004044						
7450	037304	002652							
7451	037306								
7452	037306	104404							
7453	037310	004737	004130	1\$:					
7454	037314	042737	000017	037332					
7455	037322	050537	037332						
7456	037326								
7457	037332	010000		2\$:					
7458	037334	042737	000017	037352					
7459	037342	050537	037352						
7460	037346								
7461	037352	040420		3\$:					
7462	037354								
7463	037360	061224							
7464	037362	111237	002612						
7465	037366	116104	000004						
7466	037372	123704	002612						
7467	037376	001413							
7468	037400								
7469	037416	104455							
7470	037420	000027							
7471	037422	005475							
7472	037424	010624							
7473	037426			4\$:					
7474	037426	104410							
7475	037430	000014							
7476	037432	005202							
7477	037434	005205							
7478	037436	022705	000010						

.EVEN  
ENDTST  
L10162:  
TRAP C\$ETST

BADHEAD  
:\*\*\*\*\* TEST 74 \*\*\*\*\*  
:\*ALU TEST  
:\*TEST OF ALU FUNCTION ADD W/C WITH C BIT SET  
:\*ALU FUNCTION (A PLUS B PLUS C) CODE=01  
:\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
:\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
BADHEAD  
:\*\*\*\*\* TEST 74 \*\*\*\*\*

BGNTST  
T74::  
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  
JSR PC,SETC :SET C BIT!  
BIC #17,2\$ :CLEAR ADDRESS FIELD OF INSTRUCTION  
BIS R5,2\$ :ADD ADDRESS TO INSTRUCTION  
ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304  
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  
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 IF YES  
ERROR 23 :ALU ERROR  
TRAP C\$ERDF  
.WORD 23  
.WORD EM23  
.WORD ERR23  
ESCAPE SEG  
TRAP C\$ESCAPE  
.WORD 10000\$-.  
INC R2 :NEXT DATA  
INC R5 :NEXT ADDRESS  
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						:*****			
7498						:***** TEST 75 *****			
7499						:*ALU TEST			
7500						:*TEST OF ALU FUNCTION SUB W/C WITH C BIT SET			
7501						:*ALU FUNCTION (A-B-C) CODE=2			
7502						:*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA			
7503	037464					:*PERFORM THE FUNCTION, VERIFY THE RESULTS			
7504						BADHEAD			
7505						:*****			
7506						:***** TEST 75 *****			
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 INSTRUCTION, 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											
7557	037672						.EVEN				
7558	037672						ENDTST				
7559	037672	104401					L10164:				
7560							TRAP	C\$ETST			
7561											
7562	037674										
7563							BADHEAD				
7564							:*****	TEST 76	*****		
7565							:*ALU TEST				
7566							:*TEST OF ALU FUNCTION INC A WITH C BIT SET				
7567							:*ALU FUNCTION (A PLUS 1) CODE=3				
7568							:*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA				
7569	037674						:*PERFORM THE FUNCTION, VERIFY THE RESULTS				
7570							BADHEAD				
7571							:*****	TEST 76	*****		
7572	037674										
7573	037674					BGNTST					
7574	037674					T76::					
7575	037700						MYINT				
7576	037704	005005					MSTCLR				:MASTER CLEAR M8200,4,6,7
7577	037706	012702	040072				CLR	R5			:MEM + SP ADDRESS
7578	037712	004737	003672				MOV	#5\$,R2			:POINTER TO CORRECT DATA
7579	037716	002642					JSR	PC,MEMLD			:LOAD 8 WORDS OF MAIN MEMORY
7580	037720	004737	004044				MEMDAT				:POINTER TO DATA
7581	037724	002652					JSR	PC,SPLD			:LOAD 8 WORDS OF SP
7582	037726						SPDAT				:POINTER TO DATA
7583	037726	104404					BGNSEG				
7584	037730	004737	004130			1\$:	TRAP	C\$BSEG			
7585	037734	042737	000017	037752			JSR	PC,SETC			:SET C BIT!
7586	037742	050537	037752				BIC	#17,2\$			:CLEAR ADDRESS FIELD OF INSTRUCTION
7587	037746						BIS	R5,2\$			:ADD ADDRESS TO INSTRUCTION
7588	037752	010000				2\$:	ROMCLK				:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
7589	037754	042737	000017	037772			010000				:LOAD MAR
7590	037762	050537	037772				BIC	#17,3\$			:CLEAR ADDRESS OF INSTRUCTION
							BIS	R5,3\$			:ADD ADDRESS TO INSTRUCTION





```
7647 040136          BGNSEG
7648 040136 104404   TRAP      C$BSEG
7649 040140 004737 004130   JSR      PC,SETC      ;SET C BIT!
7650 040144 042737 000017 040162   BIC     #17,2$      ;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   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   MCVB    (R2),SGDDAT  ;PUT 'EXPECTED' IN SGDDAT
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          BGNST
```

```
7703 040314
7704 040314
7705 040320
7706 040324 005005
7707 040326 012702 040512
7708 040332 004737 003672
7709 040336 002642
7710 040340 004737 004044
7711 040344 002652
7712 040346
7713 040346 104404
7714 040350 004737 004130
7715 040354 042737 000017 040372 1$:
7716 040362 050537 040372
7717 040366
7718 040372 010000
7719 040374 042737 000017 040412 2$:
7720 040402 050537 040412
7721 040406
7722 040412 040500 3$:
7723 040414
7724 040420 061224
7725 040422 111237 002612
7726 040426 116104 000004
7727 040432 123704 002612
7728 040436 001413
7729 040440
7730 040456 104455
7731 040460 000027
7732 040462 005475
7733 040464 010624
7734 040466
7735 040466 104410 4$:
7736 040470 000014
7737 040472 005202
7738 040474 005205
7739 040476 022705 000010
7740 040502 001322
7741 040504
7742 040504 10000$:
7743 040504 104405
7744 040506
7745 040506 104432
7746 040510 000012
7747 040512 001 001 000 5$:
7748 040515 000 126 126
7749 040520 253 253
7750
7751
7752 040522 .EVEN
7753 040522 ENDTST
7754 040522 104401 L10167:
7755
7756 040524
7757
7758
```

MYINT  
MSTCLR  
CLR R5 :MASTER CLEAR M8200.4.6.7  
MOV #5\$,R2 :MEM + SP ADDRESS  
JSR PC,MEMLD :POINTER TO CORRECT DATA  
MEMDAT :LOAD 8 WORDS OF MAIN MEMORY  
JSR PC,SPLD :POINTER TO DATA  
SPDAT :LOAD 8 WORDS OF SP  
BGNSEG :POINTER TO DATA  
TRAP C\$BSEG  
JSR PC,SETC :SET C BIT!  
BIC #17,2\$ :CLEAR ADDRESS FIELD OF INSTRUCTION  
BIS R5,2\$ :ADD ADDRESS TO INSTRUCTION  
ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304  
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  
040400!<4\*20> :BR A PLUS C  
ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304  
61224 :MOVE BR TO PORT4  
MOVB (R2),SGDDAT :PUT 'EXPECTED' IN SGDDAT  
MOVB 4(R1),R4 :PUT 'FOUND IN R4  
CMPB SGDDAT,R4 :DATA CORRECT?  
BEQ 4\$ :BR IF YES  
ERROR 23 :ALU ERROR  
TRAP C\$ERDF  
.WORD 23  
.WORD EM23  
.WORD ERR23  
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  
TRAP C\$ESEG  
EXIT TST  
TRAP C\$EXIT  
.WORD L10167-  
.BYTE 1,1,0,0,126,126,253,253  
C\$SETST  
BADHEAD  
:\*\*\*\*\* TEST 79 \*\*\*\*\*  
:\*ALU TEST

CZKBAO KMC11-B STATIC PART1  
CZKBA.P11 20-OCT-81 16:58

MACY11 30A(1052) 21-OCT-81 10:50 C 14 PAGE 171  
HARDWARE TESTS

SEQ 0171

7759  
7760  
7761  
7762  
7763 040524  
7764  
7765  
7766 040524  
7767 040524

;\*TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT SET  
;\*ALU FUNCTION (A-B-1) CODE=17  
;\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
;\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
BADHEAD  
;\*\*\*\*\* TEST 79 \*\*\*\*\*

BGNTST  
T79::

7768	040524					MYINT ;		
7769	040530					MSTCLR		:MASTER CLEAR M8200.4,6,7
7770	040534	005005				CLR R5		:MEM + SP ADDRESS
7771	040536	012702	040722			MOV #5\$,R2		:POINTER TO CORRECT DATA
7772	040542	004737	003672			JSR PC,MEMLD		:LOAD 8 WORDS OF MAIN MEMORY
7773	040546	002642				MEMDAT		:POINTER TO DATA
7774	040550	004737	004044			JSR PC,SPLD		:LOAD 8 WORDS OF SP
7775	040554	002652				SPDAT		:POINTER TO DATA
7776	040556					BGNSEG		
7777	040556	104404				TRAP CSBSEG		
7778	040560	004737	004130			JSR PC,SETC		:SET C BIT!
7779	040564	042737	000017	040602	1\$:	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				010000		:LOAD MAR
7783	040604	042737	000017	040622	2\$:	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),SGDDAT		:PUT 'EXPECTED' IN SGDDAT
7790	040636	116104	000004			MOVB 4(R1),R4		:PUT 'FOUND IN R4
7791	040642	123704	002612			CMPB SGDDAT,R4		:DATA CORRECT?
7792	040646	001413				BEQ 4\$		:BR IF YES
7793	040650					ERROR 23		:ALU ERROR
7794	040666	104455				TRAP CSERDF		
7795	040670	000027				.WORD 23		
7796	040672	005475				.WORD EM23		
7797	040674	010624				.WORD ERR23		
7798	040676				4\$:	ESCAPE SEG		
7799	040676	104410				TRAP CSESCAPE		
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 CSESEG		
7808	040716					EXIT TST		
7809	040716	104432				TRAP CSEXIT		
7810	040720	000012				.WORD L10170-		
7811	040722	377	000	376	5\$:	.BYTE -1,0,376,-1,-1,252,124,-1		
7812	040725	377		252				
7813	040730	124	377					
7814								
7815								
7816	040732					.EVEN		
7817	040732					ENDTST		
7818	040732	104401			L10170:	TRAP C\$ETST		
7819								
7820								
7821	040734					BADHEAD		
7822						:***** TEST 80 *****		
7823						:*ALU TEST		

```
7824
7825
7826
7827
7828 040734
7829
7830
7831 040734
7832 040734
7833 040734
7834 040740
7835 040744 005005
7836 040746 012702 041132
7837 040752 004737 003672
7838 040756 002642
7839 040760 004737 004044
7840 040764 002652
7841 040766
7842 040766 104404
7843 040770 004737 004130
7844 040774 042737 000017 041012 1$:
7845 041002 050537 041012
7846 041006
7847 041012 010000
7848 041014 042737 000017 041032 2$:
7849 041022 050537 041032
7850 041026
7851 041032 040560 3$:
7852 041034
7853 041040 061224
7854 041042 111237 002612
7855 041046 116104 000004
7856 041052 123704 002612
7857 041056 001413
7858 041060
7859 041076 104455
7860 041100 000027
7861 041102 005475
7862 041104 010624
7863 041106
7864 041106 104410 4$:
7865 041110 000014
7866 041112 005202
7867 041114 005205
7868 041116 022705 000010
7869 041122 001322
7870 041124
7871 041124
7872 041124 104405 10000$:
7873 041126
7874 041126 104432
7875 041130 000012
7876 041132 377 377 376 5$:
7877 041135 376 124 124
7878 041140 251 251
7879
```

;\*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  
BADHEAD  
:\*\*\*\*\* TEST 80 \*\*\*\*\*

BGNTST  
T80::

M: INT  
MSTCLR  
CLR R5 ;MASTER CLEAR M8200,4,6,7  
MOV #5\$,R2 ;MEM + SP ADDRESS  
JSR PC,MEMLD ;POINTER TO CORRECT DATA  
MEMDAT ;LOAD 8 WORDS OF MAIN MEMORY  
JSR PC,SPLD ;POINTER TO DATA  
SPDAT ;LOAD 8 WORDS OF SP  
BGNSEG ;POINTER TO DATA  
TRAP CSBSEG  
JSR PC,SETC ;SET C BIT!  
BIC #17,2\$ ;CLEAR ADDRESS FIELD OF INSTRUCTION  
BIS R5,2\$ ;ADD ADDRESS TO INSTRUCTION  
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304  
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  
040400! <7\*20> ;BR DEC A  
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 23 ;ALU ERROR  
TRAP CSERDF  
.WORD 23  
.WORD EM23  
.WORD ERR23  
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

CZKMBAO KMC11-8 STATIC PART1  
CZKMB.A.P11 20-OCT-81 16:58

MACY11 30A(1052) 21-OCT-81 10:50 F 14 PAGE 174  
HARDWARE TESTS

SEQ 0174

7880  
7881 041142  
7882 041142  
7883 041142 104401  
7884  
7885

.EVEN  
ENDTST  
L10171:  
TRAP CSETST

CZKMBAO KMC11-B STATIC PART1  
CZKMB.A.P11 20-OCT-81 16:58

MACY11 30A(1052) 21-OCT-81 10:50 <sup>6 14</sup> PAGE 175  
HARDWARE TESTS

SEQ 0175

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.  
////////////////////////////////////

7892  
7893  
7894  
7895  
7896  
7897  
7898  
7899  
7900  
7901  
7902  
7903  
7904  
7905 041144  
7906 041144 000015  
7907 041146  
7908  
7909  
7910 041146  
7911 041146 001031  
7912 041150 041267  
7913 041152 160000  
7914 041154 177776  
7915 041156  
7916 041156 002031  
7917 041160 041306  
7918 041162 000000  
7919 041164 000674  
7920 041166  
7921 041166 003032  
7922 041170 041330  
7923 041172 007000  
7924 041174 000004  
7925 041176 000007  
7926  
7927  
7928  
7929  
7930  
7931 041200  
7932  
7933 041200  
7934  
7935 041200 044127 041511 020110  
7936 041206 044515 051103 026517  
7937 041214 051120 041517 051505  
7938 041222 047523 035122 000  
7939 041227 060 046475 031070  
7940 041234 030060 032054 046475  
7941 041242 031070 032060 033054  
7942 041250 046475 031070 033060  
7943 041256 033454 046475 031070  
7944 041264 033460 000  
7945 041267 103 051123 040440  
7946 041274 042104 042522 051523  
7947 041302 035040 000040

BGNHRD  
.WORD L10172-LSHARD/2  
LSHARD::  
:  
GPRMD WMP,0,0,7,0,7,YES  
GPRMA ADDRES,2,0,160000,177776,YES  
.WORD TSCODE  
.WORD ADDRES  
.WORD TSLOLIM  
.WORD TSHILIM  
GPRMA VECTOR,4,0,0,674,YES  
.WORD TSCODE  
.WORD VECTOR  
.WORD TSLOLIM  
.WORD TSHILIM  
GPRMD PIRTY,6,0,7000,4,7,YES  
.WORD TSCODE  
.WORD PIRTY  
.WORD 7000  
.WORD TSLOLIM  
.WORD TSHILIM  
:  
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  
ENDHRD  
.EVEN  
L10172:  
WMP: .ASCIZ 'WHICH MICRO-PROCESSOR:'  
:  
.ASCIZ '0=M8200,4=M8204,6=M8206,7=M8207'  
:  
ADDRES: .ASCIZ /CSR ADDRESS : /



7948	041306	042526	052103	051117	VECTOR: .ASCIZ /VECTOR ADDRESS : /
7949	041314	040440	042104	042522	
7950	041322	051523	035040	000040	
7951	041330	051120	047511	044522	PRIPTY: .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					

7992  
7993  
7994  
7995  
7996  
7997  
7998  
7999  
8000  
8001  
8002  
8003  
8004  
8005  
8006  
8007  
8008  
8009  
8010  
8011  
8012  
8013  
8014  
8015  
8016  
8017  
8018  
8019  
8020  
8021  
8022  
8023  
8024  
8025  
8026  
8027  
8028  
8029  
8030  
8031  
8032  
8033  
8034  
8035  
8036  
8037  
8038  
8039  
8040  
8041  
8042  
8043  
8044  
8045  
8046

041626  
041626 000000  
041630  
  
041630  
041630  
  
  
  
041630  
  
041630  
041630 000000  
041732 041732  
041734 000000  
  
041734 000000  
041736 000000  
041740  
  
000120  
  
  
  
  
  
  
  
  
  
000001

.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.

BGNSFT  
.WORD L10173-L\$SOFT/2  
L\$SOFT::

ENDSFT  
.EVEN  
L10173:  
.EVEN

ENDMOD

CORMAX:  
.WORD 0 ;START OF NPR AREA (TEST 55)  
MEMEND: .WORD 0 ;END OF NPR AREA  
LASTAD  
.EVEN  
.WORD 0  
L\$LAST:: .WORD 0

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!

.END



CSDODU= 000051	691#													
CSDRPT= 000024	691#													
CSDU = 000053	691#	2751												
CSEDI1= 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	







FSCLEA= 0000C7  
FSDU = 000016  
FSEND = 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								
691#	7906	7932										
691#	922	937										
691#	2572	2709										
691#	2541	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	
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\$HARD= 000004  
F\$HW = 000013  
F\$INIT= 000016  
F\$JMP = 000050

F\$MOD = 000000  
F\$MSG = 000011



FSPROT= 000021  
FSPWR = 000017  
FSRPT = 000012  
FSSEG = 000003

691#	2550	2555												
691#														
691#	2537	2546												
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														

FSSOFT= 000005  
FSSRV = 000010  
FSSUB = 000002  
FSSW = 000014  
FSTEST= 000001

691#	8005	8010												
691#														
691#	4990	5062												
691#	951	957												
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										
2614	2623#	2630												

GETPRM 012626  
GSCNTO= 000200  
GSDLM= 000372  
GSDISP= 000003  
GSEXCP= 000400  
GSHILI= 000002  
GSLOLI= 000001  
GSNO = 000000  
GSOFFS= 000400  
GSOFSI= 000376  
GSPRMA= 000001  
GSPRMD= 000002  
GSPRML= 000000  
GSRADA= 000140  
GSRADB= 000000  
GSRADD= 000040  
GSRADL= 000120  
GSRADO= 000020  
GSXFER= 000004  
GSYES = 000010  
HELP = 000000

691#														
691#	1531	4783	4796	4805	4814	4823	4832	4841						
691#														
691#														
691#														
691#														
691#	7911	7916	7921											
691#	7911	7916	7921											
691#	7911	7916												
691#	7921													
691#														
691#														
691#														
691#														
691#	7911	7916	7921											
691#														
691#	7911	7916	7921											
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#													
ISAU = 000041	691#	2766#	2770#											
ISAUTO= 000041	691#	2557#	2561#											
ISCLN = 000041	691#	2726#	2732#											
ISDU = 000041	691#	2745#	2752#											
ISHRD = 000041	7906#	7934#												
ISINIT= 000041	691#	2572#	2711#											
ISMOD = 000041	691#	697#	8020#											
ISMSG = 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#	
ISPRUT= 000040	691#	2550#												
ISPTAB= 000041	691#													
ISPR = 000041	691#													
ISRPT = 000041	691#	2537#	2548#											
ISSEG = 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#	7799	7808#	7832	7842#	
	7864	7873#												
ISSETU= 000041	691#													
ISSFT = 000041	8005#	8012#												
ISSRV = 000041	691#													
ISSUB = 000041	691#	2789	2825	2861	2925	2974	3025	3073	3120	3171	3235	3300	3362	





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#	



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															
SETBRO 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						



SVCINS= 000000

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

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	251R	2519	2520	2521	2522	2523	2524	2525	2526	2527	2528	2541
2542	254J	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	3247	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

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	4521	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

6492	6493	6494	6495	6496	6497	6498	6504	6505	6506	6507	6508	6516
6517	6540	6541	6557	6558	6559	6560	6561	6552	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
6687	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

SVCSUB= 000000  
SVCTAG= 000000

SVCTST= 000000

SWPAC1 041443  
SWPAC2 041504  
SSL SYM= 010000

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	5515	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												
691#	702#	2789	2790	2825	2826	2861	2862	2925	2926	2974	2975	3025
3025	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							
7964#												
7970#												
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#	2541#	2560#	2710#	2731#
2751#	2769#	2817#	2839#	2867#	2893#	2915#	2930#	2964#	2971#	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#						
5772	5774	5804#										
1086#												
1299*	1305#											
1563#	1897	1922	1947	1972	1997	2228						
1568#	1824	1848	1872	2021	2045	2086	2169	2269	2293	2342	2391	2449
2473	2497	2521										
1574#	2111	2318	2367									
1571#												
5579	5587#											
5556	5586#											
1119#	2696*	2701*	2706*									
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#

TDATA 030532  
TEMP 002416  
TESTAD 003224  
TFM1 004277  
TFM2 004330  
  
TFM27 004362  
TFM5 004345  
TOUTP 027552  
TOUTT 027546  
TYPE 002602  
T\$ARGC= 000001



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									
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#											
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#							
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#												
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#

TSCODE= 003032  
TSERRN= 000027

TSEXCP= 000000  
TSFLAG= 000040

TSGMAN= 000000  
TSHILI= 000007  
TSLAST= 000001  
TSLOLI= 000004  
TSLSYM= 010000

TSLTNO= 000120  
TSNEST= 177777

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#									
697#	8020											
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	
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\$NSO = 000000  
T\$NS1 = 000005

T\$NS2 = 000003

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#	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	5176#	5197	5199#
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

TSNS3 = 000003  
TSPTNU= 000000  
TSSAVL= 177777  
TSSEGL= 177777

TSSEKO= 010000



TSSUBN= 000000

TSTAGL= 177777  
TSTAGN= 010174

TSTEMP= 000000

7387#	7410	7416	7452#	7475	7481	7518#	7541	7547	7583#	7606	7612	7648#
7671	7677	7713#	7736	7742	7777#	7800	7806	7842#	7865	7871		
691#	2789#	2825#	2801#	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#									
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#					
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#	937#	957#	1829#	1853#	1877#	1902#	1927#	1952#	1977#	2002#
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#

TSTEST= 000120

TSTSTM= 177777

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#		
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				
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	6431	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	7855	7864	7872	7874	7883			
691#	2790#	2826#	2862#	2926#	2975#	3026#	3074#	3121#	3172#	3236#	3301#	3363#
3418#	3452#	3525#	3601#	3673#	3745#	3817#	3889#	3965#	4043#	415#	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#										
2766#	2768											
2557#	2559											
2726#	2730											
2745#	2750											
7906#	7933											
922#	937											
2572#	2709											
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	
2550#												
2537#	2541	2546										
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#

TSTSTS= 000001

TSSAU = 010050  
TSSAUT= 010044  
TSSCLE= 010046  
TSSDU = 010047  
TSSHAR= 010172  
TSSHW = 010000  
TSSINI= 010045  
TSSMSG= 010041

TSSPRO= 010043  
TSSRPT= 010042  
TSSSEG= 010000

TSS\$OF= 010173  
TSS\$SUB= 010113  
TSS\$W = 010001  
TSS\$TES= 010171

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#		
8005#	8011											
4990#	5062											
951#	957											
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			
826	2789#											
835	3235#											
836	3300#											
837	3362#											
838	3417#											
839	3451#											
840	3524#											
841	3600#											
842	3672#											
843	3744#											
844	3816#											
827	2825#											
845	3888#											
846	3964#											
847	4042#											
848	4114#											
849	4186#											
850	4258#											
851	4330#											
852	4402#											
853	4474#											
854	4546#											
828	2861#											
855	4618#											
856	4700#											
857	4775#											
858	4893#											
859	4983#											
4989#												

T1	013176	G
T10	014770	G
T11	015174	G
T12	015376	G
T13	015562	G
T14	015674	G
T15	016134	G
T16	016430	G
T17	016670	G
T18	017130	G
T19	017370	G
T2	013320	G
T20	017630	G
T21	020120	G
T22	020422	G
T23	020662	G
T24	021122	G
T25	021362	G
T26	021622	G
T27	022062	G
T28	022322	G
T29	022562	G
T3	013366	G
T30	023022	G
T31	023330	G
T32	023726	G
T33	024412	G
T34	024722	G
T34.1	024740	

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	

XSALWA= 000000  
XSFALS= 000040  
X\$OFFS= 000400  
X\$TRUE= 000020  
ZERU 002552  
\$BDADR 0C2610

691#														
691#														
691#														
691#														
1106#														
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*		
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*	5419*	5465*	5531*	5581*	5625*	5639*	5681*	5726*		
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

\$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								