

KMC11-B

KMC11-B STATIC PART 2 AH-S880A-MC
CZKMCA0 FICHE 1 OF 1

APR 1982
COPYRIGHT © 1982
MADE IN USA



A large grid of small tables, each containing technical data. The tables are arranged in approximately 15 rows and 15 columns. Each individual table contains multiple columns of data, including numerical values, alphanumeric codes, and possibly small diagrams or symbols. The data is organized in a structured, tabular format, typical of a technical manual or data sheet. The text within the tables is small and difficult to read individually, but the overall layout is consistent across the entire page.

.NLIST SEQ,BIN,LOC
.REM &

IDENTIFICATION

PRODUCT CODE: AC-S878A-MC
PRODUCT NAME: CZKMCAO KMC11-B STATIC PART2
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 THE LOGIC OF THE KMC11-B. IT IS RUN AFTER, AND IN CONJUNCTION WITH, CZKMB 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

CZKMB KMC11-B STATIC PART1

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>

RUN REMOTE POWERFAIL TEST ? 0=NO, 1=YES : (0) 0 ? <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

IF THE SYSTEM IS CONFIGURED TO BOOT ON POWER FAIL, DO NOT
RUN THE REMOTE POWER FAIL TEST.

2.3 PROGRAM OPTIONS

2.3.1 START COMMAND

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

2.3.1.1 TESTS SWITCH (/TESTS:<TEST-LIST>)

<TEST-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS. ON THIS AND ALL SWITCHES, THE ANGLE BRACKETS <> ARE PUNCTUATION USED IN THE DEFINITION ONLY, AND ARE NOT TO BE TYPED BY THE OPERATOR. SEE EXAMPLE AT END OF 2.3.1.

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

<PASS-CNT> IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING EXECUTION. IN THIS CASE EXIT FROM THE PROGRAM IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C OR BY OCCURANCE OF AN ERROR WITH THE HALT ON ERROR FLAG BEING SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF 2.3.1.

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

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

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

IDR INHIBIT DROPPING OF UNITS BY DIAGNOSTIC

THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED. SEE EXAMPLE AT END OF 2.3.1.

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>.0 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 (0) 174100 ?
RUN REMOTE POWERFAIL TEST ? 0=NO, 1=YES 0?

*** NOTE: PI VECTOR IS NOT NEEDED FOR THIS TEST ***

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 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 85 SECONDS.

3.0 ERROR INFORMATION

3.1 ERROR REPORTING

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

CZKMCA 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 CUMULATIVE ERROR COUNT IS GIVEN AFTER PASS.

4.2 PROGRESS REPORTS

A PASS COUNT IS UPDATED AND PRINTED AFTER PASS.

5.0 DEVICE INFORMATION TABLES

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

6.0 TEST SUMMARIES

TEST SUMMARIES ARE GIVEN AT THE BEGINNING OF EACH
TEST MODULE.

672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703

002000

.TITLE CZKMCAO KMC11-B STATIC PART2
.=2000

002000

.MCALL SVC
SVC

; INITIALIZE SUPERVISOR MACROS

002000

BGNMOD CZKMCA

000000
000000
000000
000000
000000
000000
000000

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

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

704
705
706
707
708
709
710 002000
711
712
713 002000
714 002000
715 002000 103
716 002001 132
717 002002 113
718 002003 115
719 002004 103
720 002005 000
721 002006 000
722 002007 000
723 002010
724 002010 101
725 002011
726 002011 060
727 002012
728 002012 000000
729 002014
730 002014 000360
731 002016
732 002016 031256
733 002020
734 002020 000000
735 002022
736 002022 002252
737 002024
738 002024 002300
739 002026
740 002026 032002
741 002030
742 002030 000000
743 002032
744 002032 000000
745 002034
746 002034 000000
747 002036
748 002036 000000
749 002040
750 002040 002124
751 002042
752 002042 000000
753 002044
754 002044 000000
755 002046
756 002046 000000
757 002050
758 002050 003
759 002051 003

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

        POINTER BGNAU,BGNDU,BGNRPT,BGNSW

        HEADER CZKMC,A,0,240.,0
L$NAME:: :DIAGNOSTIC NAME
        .ASCII /C/
        .ASCII /Z/
        .ASCII /K/
        .ASCII /M/
        .ASCII /C/
        .BYTE 0
        .BYTE 0
        .BYTE 0
L$REV:: :REVISION LEVEL
        .ASCII /A/
L$DEPO:: :0
        .ASCII /0/
L$UNIT:: :NUMBER OF UNITS
        .WORD 0
L$TIML:: :LONGEST TEST TIME
        .WORD 240.
L$HPCP:: :POINTER TO H.W. QUES.
        .WORD L$HARD
L$SPCP:: :POINTER TO S.W. QUES.
        .WORD 0
L$HPTP:: :PTR. TO DEF. H.W. PTABLE
        .WORD L$HW
L$SPTP:: :PTR. TO S.W. PTABLE
        .WORD L$SW
L$LADP:: :DIAG. END ADDRESS
        .WORD L$LAST
L$STA:: :RESERVED FOR APT STATS
        .WORD 0
L$CO:: :
        .WORD 0
L$DTYP:: :DIAGNOSTIC TYPE
        .WORD 0
L$APT:: :APT EXPANSION
        .WORD 0
L$DTP:: :PTR. TO DISPATCH TABLE
        .WORD L$DISPATCH
L$PRIO:: :DIAGNOSTIC RUN PRIORITY
        .WORD 0
L$ENVI:: :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
```


760	002052		LSEF::			;DIAG. EVENT FLAGS
761	002052	000000		.WORD	0	
762	002054	000000		.WORD	0	
763	002056		LSSPC::			
764	002056	000000		.WORD	0	
765	002060		LSDEVP::			; POINTER TO DEVICE TYPE LIST
766	002060	002670		.WORD	LSDVTYP	
767	002062		LSREPP::			;PTR. TO REPORT CODE
768	002062	012366		.WORD	LSRPT	
769	002064		LSEXP4::			
770	002064	000000		.WORD	0	
771	002066		LSEXP5::			
772	002066	000000		.WORD	0	
773	002070		LSAUT::			;PTR. TO ADD UNIT CODE
774	002070	013114		.WORD	LSAU	
775	002072		LSDUT::			;PTR. TO DROP UNIT CODE
776	002072	013110		.WORD	LSDU	
777	002074		LSLUN::			;LUN FOR EXERCISERS TO FILL
778	002074	000000		.WORD	0	
779	002076		LSDESP::			;POINTER TO DIAG. DESCRIPTION
780	002076	002676		.WORD	LSDESC	
781	002100		LSLOAD::			;GENERATE SPECIAL AUTOLOAD EMT
782	002100	104035		EMT	ESLOAD	
783	002102		LSETP::			;POINTER TO ERR_TBL
784	002102	000000		.WORD	0	
785	002104		LSICP::			;PTR. TO INIT CODE
786	002104	012404		.WORD	LSINIT	
787	002106		LSCCP::			;PTR. TO CLEAN-UP CODE
788	002106	013106		.WORD	LSCLEAN	
789	002110		LSACP::			;PTR. TO AUTO CODE
790	002110	012402		.WORD	LSAUTO	
791	002112		LSPRT::			;PTR. TO PROTECT TABLE
792	002112	012374		.WORD	LSPROT	
793	002114		LSTEST::			;TEST NUMBER
794	002114	000000		.WORD	0	
795	002116		LSDLY::			;DELAY COUNT
796	002116	000000		.WORD	0	
797	002120		LSHIME::			;PTR. TO HIGH MEM
798	002120	000000		.WORD	0	
799						
800						
801						
802						
803						
804						
805						

806
807
808
809
810
811
812
813 002122
814 002122 000052
815 002124
816 002124 013116
817 002126 013226
818 002130 013420
819 002132 013572
820 002134 013750
821 002136 014230
822 002140 014460
823 002142 014720
824 002144 015270
825 002146 015650
826 002150 016134
827 002152 016420
828 002154 016664
829 002156 017144
830 002160 017424
831 002162 017704
832 002164 020164
833 002166 020444
834 002170 020724
835 002172 021210
836 002174 021474
837 002176 021760
838 002200 022244
839 002202 022530
840 002204 023014
841 002206 023222
842 002210 023404
843 002212 023744
844 002214 024174
845 002216 024440
846 002220 024742
847 002222 025344
848 002224 026442
849 002226 026612
850 002230 027002
851 002232 027502
852 002234 027676
853 002236 030026
854 002240 030244
855 002242 030362
856 002244 030522
857 002246 030660
858
859
860
861

.SBTTL DISPATCH TABLE

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

DISPATCH 42
.WORD 42
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

CZKMAO KMC11-B STATIC PART2
CZKMA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 F 2
DISPATCH TABLE PAGE 18

SEQ 0018

862
863

864
865
866
867
868
869
870
871
872
873 002250
874 002250 000012
875 002252
876 002252
877 002252 000006
878 002254 174100
879 002256 000300
880 002260 005000
881 002262 000003
882 002264 000056
883 002266 000000
884 002270 000000
885 002272 000000
886 002274 000000
887
888 002276
889 002276

.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.
://

.ENABL AMA
BGNHW DFPTBL
.WORD L10000-L\$HW/2

L\$HW::
DFPTBL::

.WORD 6
.WORD 174100
.WORD 300
.WORD 5000
.WORD 3
.WORD 56
.WORD 0
.WORD 0
.WORD 0
.WORD 0

:MICRO-CPU TYPE.
:M8200,4,6,7 CRS ADDRESS
:M8200,4,6,7 VECTOR ADDRESS
:INTERRUPT PRIORITY LEVEL
:LINE UNIT TYPE
:SWITCH PACK #1 (DDCMP LINE #)
:SWITCH PACK #2 (BM873 BOOT ADDRESS)
:SWITCH PACK #3
:TEST CONNECTOR INSTALLED FLAG
:TEST REMOTE POWERFAIL FLAG (0=NO, 1=YES)

ENDHW
L10000:

890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910

002276
002276 000000
002300
002300

002300
002300

```
.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:
```

911
912
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
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966

002300

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

000040
000037

.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

967 000036
968 000035
969 000034
970
971
972
973
974 000340
975 000300
976 000240
977 000200
978 000140
979 000100
980 000040
981 000000
982
983
984
985 000004
986 000010
987 000020
988 000040
989 000100
990 000200
991 000400
992 001000
993 002000
994 004000
995 010000
996 020000
997 040000
998 100000

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

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

...
: 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

;KMC11-B DEFINITIONS

999			
1000			
1001	000000	IMM==	000000
1002	020000	IBUS==	020000
1003	040000	MEM==	040000
1004	060000	BRG==	060000
1005	100000	BRIM==	100000
1006	120000	IBSP==	120000
1007	140000	BRMEM==	140000
1008	160000	BRBRG==	160000
1009			
1010	004000	LMH==	004000
1011	010000	LML==	010000
1012	014000	IMR==	014000
1013	010000	A9==	LML
1014	004000	A8==	LMH
1015			
1016	000400	BRGR==	000400
1017	001000	OBP==	001000
1018	001400	BRS==	001400
1019	002000	OB==	002000
1020	002400	M==	002400
1021	003000	SPR==	003000
1022	003400	SPBR==	003400
1023			
1024	000400	BRA==	000400
1025	001000	C1==	001000
1026	001400	Z1==	001400
1027	002000	BRG0==	002000
1028	002400	BRG1==	002400
1029	003000	BRG4==	003000
1030	003400	BRG7==	003400
1031			
1032	000200	SELA==	000200
1033	000220	SELB==	000220
1034			
1035	000000	SRG0==	000000
1036	000020	SRG1==	000020
1037	000040	SRG2==	000040
1038	000060	SRG3==	000060
1039	000100	SRG4==	000100
1040	000120	SRG5==	000120
1041	000140	SRG6==	000140
1042	000160	SRG7==	000160
1043	000200	SRG10==	000200
1044	000220	SRG11==	000220
1045	000240	SRG12==	000240
1046	000260	SRG13==	000260
1047	000300	SRG14==	000300
1048	000320	SRG15==	000320
1049	000340	SRG16==	000340
1050	000360	SRG17==	000360
1051			

CZKMCAO KMC11-B STATIC PART2
CZKMCA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 L² PAGE 24
GLOBAL EQUATES SECTION

SEQ 0024

1052
1053
1054
1055
1056
1057

::*****
::* PROGRAM EVENT FLAG DEFINITIONS
::*****

1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068 002300 000000
1069 002302 000000
1070
1071
1072
1073
1074 002304 000000
1075 002306 000000
1076 002310 000000
1077 002312 000000
1078 002314 000000
1079 002316 000000
1080 002320 000000
1081 002322 000000
1082 002324 000000
1083 002326 000000
1084 002330 000000
1085 002332 000000
1086 002334 000001
1087 002336 000000
1088 002340 000001
1089 002342 000001
1090 002344 000001
1091 002346 000001
1092 002350 000000
1093 002352 000000
1094 002354 000000
1095 002356 000006
1096 002360 000000
1097 002362 000000
1098 002364 000000
1099 002366 000000
1100 002370 000000
1101 002372 000000
1102 002374 000000
1103 002376 000000
1104 002400 003777
1105 002402 000000
1106 002404 000000
1107 002406 000000
1108 002410 000000
1109 002412 000000
1110 002414 000000
1111 002416 000000
1112 002420 000000
1113 002422 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

:*****
:* MISCELLANEOUS 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 :KM 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 0 :HIGHEST LOCATION FOR NPR'S
KMACTV: .BLKW 1 :M8200,4,6,7 SELECTED ACTIVE
KMNUM: .BLKW 1 :OCTAL NUMBER OF M8200,4,6,7'S
SAVACT: .BLKW 1 :ORIGINAL ACTIVE DEVICES
SAVNUM: .BLKW 1 :WORKABLE NUMBER
FLAG: .WORD 0 :SCRATCH STORAGE
RUN: .WORD 0 :POINTER TO RUNNING DEVICES
FADR: .WORD 0
WTYPE: .WORD 6 :M82XX NUMBER FOR TYPE OF MICO-CPU
\$REG5: .WORD 0 :STORAGE USED FOR ERROR MSG DATA
\$REG4: .WORD 0
\$REG3: .WORD 0
\$REG2: .WORD 0
\$REG1: .WORD 0
\$REG0: .WORD 0
TYPE: .WORD 0 :=0 FOR DMP,=1 FOR M8206
MRO: .WORD 0 :MEMLOC USED INSTEAD OF RO.
MEMSZ: .WORD 3777 :INDICATES MEMORIE SIZE, LAST ADDR.
TEMP: .WORD 0
STEMPO: .WORD 0
STMP0: .WORD 0
\$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
 .WORD 0

1114 002424 000000
1115 002426 000000
1116 002430 000000
1117 002432 000000

FTIME: .WORD 0
SAVE4: .WORD 0
SAVE6: .WORD 0
SKIP33: .WORD 0 ; FLAG FOR TESTING REMOTE POWERFAIL (T33)

1118
1119
1120
1121
1122 002434 000
1123 002436
1124 002436 000
1125 002437 000
1126
1127

;* 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

1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146

;* 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

1147 002440 000000
1148 002442 000000
1149 002444 000000
1150

STAT1: .WORD 0
STAT2: .WORD 0
STAT3: .WORD 0

1151
1152
1153
1154 002446 000000
1155 002450 000000
1156 002452 000000
1157 002454 000000
1158 002456 000000
1159 002460 000000
1160 002462 000000
1161 002464 000000
1162 002466 000000
1163

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

KMRVEC: 0 ;POINTER TO M8200,4,6,7 RCV INTRPT VECTOR
KMRLVL: 0 ;POINTER TO M8200,4,6,7 RCV INTRPT SERVICE PS
KMTVEC: 0 ;POINTER TO M8200,4,6,7 TX INTRPT VECTOR
KMTLVL: 0 ;POINTER TO M8200,4,6,7 TX INTRPT SERVICE PS
KMCSR: 0 ;POINTER TO M8200,4,6,7 CONTROL STATUS REGISTER
KMCSRH: 0 ;POINTER TO M8200,4,6,7 CONTROL STATUS REGISTER HIGH BYTE
KMCTL: 0 ;POINTER TO M8200,4,6,7 CONTROL OUT REGISTER
KMPO4: 0 ;POINTER TO M8200,4,6,7 PORT REGISTER - SEL4
KMPO6: 0 ;POINTER TO M8200,4,6,7 PORT REGISTER - SEL6

1164
1165
1166 002470
1167

***** PRIMARY REG ADRS STORAGE FOR THIS UNIT *****
;THESE LOCATIONS WILL BE LOADED FOR THE CURRENT UNIT, IN INIT CODE
REGADR:

1168
1169 002470 000100

***** STACK USED FOR SUBROUTINE LINKAGE *****
.BLKW 100

CZKMCAO KMC11-B STATIC PART2
CZKMCA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 ^{B 3} PAGE 27
GLOBAL DATA SECTION

SEQ 0027

1170 002670
1171
1172
1173
1174
1175
1176
1177

SSTACK:

1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211

002670				
002670				
002670	034115	030062	000066	
002676				
002676				
002676	055103	046513	040503	
002704	020060	046513	030503	
002712	026461	020102	052123	
002720	052101	041511	050040	
002726	051101	031124	000	
002734				

```

.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
:*****
DEVTYP <M8206>
LSDVTYP::
.ASCIZ /M8206/
.EVEN
DESCRIPT <CZKMCAO KMC11-B STATIC PART2>
LSDESC::
.ASCIZ /CZKMCAO KMC11-B STATIC PART2/
.EVEN

:
: FORMAT STATEMENTS USED IN PRINT CALLS
:

```

1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226

.SBTTL GLOBAL SUBROUTINES

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

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

.MACRO POPSP2
22626
.ENDM

CZKMCAO KMC11-B STATIC PART2
CZKMCA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 E 3
GLOBAL SUBROUTINES PAGE 30

SEQ 0030

1227
1228
1229
1230
1231

1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252

```
:/
: / THE GLOBAL SUBROUTINES ARE CALLED BY MORE THAN ONE TEST
: /
: /
```

```
:-----
: MACRO'S NEEDED TO CALL SUBROUTINES
:-----
```

```
.MACRO ED$CALL XY
.LIST
:***** TEST 'XY' *****
.NLIST
.ENDM
.MACRO BADHEAD
.RADIX 10
ED$CALL \T$TESTNUM+1
.RADIX 8
.ENDM
```

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
 1296
 1297
 1298
 1299
 1300
 1301
 1302
 1303
 1304
 1305
 1306
 1307
 1308

```

.MACRO MYINT
JSR R5,DEBUG
.ENDM

.MACRO MACEX ?N2
TST TYPE
BNE N2
EXIT TST
N2:
.ENDM
.MACRO MACEX2 ?N2
CMP WTYPE,#0
BNE N2
EXIT TST
N2:
.ENDM
.MACRO K4ONLY ?N2
CMP MEMSZ,#2000
BNE N2
EXIT TST
N2:
.ENDM
;NOTE THIS TEST IS ONLY DESIGNED FOR 4K MODULE.

.MACRO CLRMAR
ROMCLK
004000
.ENDM
.MACRO ROMCLK
JSR R5,.ROMCLK
.ENDM
;CLOCK INSTRUCTION

.MACRO SROMCLK
JSR R5,.SROMCLK
.ENDM

.MACRO SKIP06 NNN
CMP WTYPE,#6
BEQ NNN
.ENDM
;SEE IF M8206

.MACRO SKIP07 NNN
CMP WTYPE,#7
BEQ NNN
.ENDM
;SEE IF M8200,4,6,7

.MACRO SKIP04 NNN
CMP WTYPE,#4
BEQ NNN
.ENDM
;SEE IF M8204

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

.MACRO CHKREG ?N2,?N3,A,B,C,D,E,F,G,H,I
.LIST ME
JSR PC,RDKM ; READ KMC11 REGISTERS
MOV #N2,R2 ; LOAD POINTER
JSR PC,RCHECK ; CHECK FOR NO CHANGE

```

CZKMAO KMC11-B STATIC PART2
CZKMA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 H 3 PAGE 33
GLOBAL SUBROUTINES

SEQ 0033

1309
1310
1311
1312
1313
1314

N2: BR N3
N3: .BYTE A,B,C,D,E,F,G,H,I,20 : SKIP LIST
.ENDM .NLIST ME
.NLIST ME


```

1315
1316
1317 002734 010446
1318 002736 012504
1319 002740 005114
1320 002742 001407
1321 002744 005114
1322 002746 012437 002756
1323 002752
1324 002756 000400
1325 002760 000767
1326 002762 005114
1327 002764 012604
1328 002766 000205
1329
1330 002770 004737 003750
1331 002774 004537 002734
1332 003000 003172
1333 003002 105077 177450
1334 003006 005077 177450
1335 003012 000207
1336
1337 003014 004737 003032
1338 003020 004737 003056
1339 003024 004737 003102
1340 003030 000207
1341
1342 003032 004537 002734
1343 003036 003222
1344 003040 017737 177416 003160
1345 003046 017737 177412 003162
1346 003054 000207
1347
1348 003056 004537 002734
1349 003062 003234
1350 003064 017737 177372 003164
1351 003072 017737 177366 003166
1352 003100 000207
1353
1354 003102 004537 002734
1355 003106 003246
1356 003110 017737 177350 003170
1357 003116 000207
1358
1359 003120 012701 003160
1360 003124 012703 000012
1361 003130 112105
1362 003132 112204
1363 003134 042705 177400
1364 003140 042704 177400
1365 003144 120504
1366 003146 001401
1367 003150 000240
1368 003152 005303
1369 003154 001365
1370 003156 000207

;TEST SUBROUTINES
SINGLE: MOV R4, -(SP) ; SAVE R4
        MOV (R5)+, R4 ; GET PARAMETER LIST
1$: COM (R4) ; CHECK FOR EXIT
   BEQ 3$ ; SKIP IF ENTRY=-1
   COM (R4) ; RESTORE VALUE
   MOV (R4)+, 2$ ; FETCH FOR EXEC.
   ROMCLK
2$: 400 ; KMC INST. GOES HERE
   BR 1$ ; GET NEXT
3$: COM (R4) ; RESTORE TERMINATOR
   MOV (SP)+, R4 ; RESTORE
   RTS R5 ; EXIT

CLRKM: JSR PC, CLRALL ; CLEAR BR, C, AND Z
        JSR R5, SINGLE ; EXECUTE CLR REGS INST.
        CLR LST ; LIST OF INSTRUCTIONS
        CLRB @KMCSR ; CLEAR BSELO
        CLR @KMCTL ; CLEAR SEL2
        RTS PC ; EXIT

RDKM: JSR PC, RDIB03 ; READ IBUS0-3
        JSR PC, RDIB47 ; READ IBUS4-7
        JSR PC, RDIP10 ; READ UBUS*10-11
        RTS PC ; EXIT

RDIB03: JSR R5, SINGLE ; READ IBUS0-3
        MOV IB03LS @KMCTL, SIB0 ; GET IB0 IB1
        MOV @KMPO4, SIB2 ; GET IB2 IB3
        RTS PC ; EXIT

RDIB47: JSR R5, SINGLE ; READ IBUS 4-7
        MOV IB47LS @KMCTL, SIB4 ; GET IB4 IB5
        MOV @KMPO4, SIB6 ; GET IB6 IB7
        RTS PC ; EXIT

RDIP10: JSR R5, SINGLE ; READ IBUS* 10-11
        MOV IP10LS @KMPO4, SIBP10 ; GET IB*10 AND 11
        RTS PC

RCHECK: MOV #SIB0, R1 ; POINTER TO EXPECTED
        MOV #10, R3 ; COUNT
RCLP: MOVB (R1)+, R5 ; EXPECTED
        MOV (R2)+, R4 ; ACTUAL
        BIC #177400, R5 ; CLEAR HIGH BYTE
        BIC #177400, R4
        CMPB R5, R4 ; CHECK DATA
        BEQ RCXX ; SKIP IF OK
        NOP ; REPLACE WITH HALT FOR CROSSTALK CHECK
RCXX: DEC R3 ; DONE?
        BNE RCLP ; NO
        RTS PC ; EXIT

```

```
1371 ; STORAGE FOR REGISTERS READ
1372
1373 003160 000 SIB0: .BYTE 0
1374 003161 000 SIB1: .BYTE 0
1375 003162 000 SIB2: .BYTE 0
1376 003163 000 SIB3: .BYTE 0
1377 003164 000 SIB4: .BYTE 0
1378 003165 000 SIB5: .BYTE 0
1379 003166 000 SIB6: .BYTE 0
1380 003167 000 SIB7: .BYTE 0
1381
1382 003170 000 SIBP10: .BYTE 0
1383 003171 000 SIBP11: .BYTE 0
1384
1385 .EVEN
1386
1387 ; COMMAND LISTS
1388
1389 003172 062220 CLRLST: BRG!OB!SELB!0 ; MOVE BRG TO IBUS 0-7
1390 003174 062221 BRG!OB!SELB!1
1391 003176 062222 BRG!OB!SELB!2
1392 003200 062223 BRG!OB!SELB!3
1393 003202 062224 BRG!OB!SELB!4
1394 003204 062225 BRG!OB!SELB!5
1395 003206 062226 BRG!OB!SELB!6
1396 003210 062227 BRG!OB!SELB!7
1397 003212 061230 BRG!OBP!SELB!10 ; MOVE BRG TO OBUS*10,11,13
1398 003214 061231 BRG!OBP!SELB!11
1399 003216 061233 BRG!OBP!SELB!13
1400 003220 177777 -1
1401
1402 003222 021002 IB03LS: IBUS!OBP!SRG0!2
1403 003224 021023 IBUS!OBP!SRG1!3
1404 003226 021044 IBUS!OBP!SRG2!4
1405 003230 021065 IBUS!OBP!SRG3!5
1406 003232 177777 -1
1407
1408 003234 021102 IB47LS: IBUS!OBP!SRG4!2
1409 003236 021123 IBUS!OBP!SRG5!3
1410 003240 021144 IBUS!OBP!SRG6!4
1411 003242 021165 IBUS!OBP!SRG7!5
1412 003244 177777 -1
1413
1414 003246 121204 IP10LS: IBSP!OBP!SRG10!4
1415 003250 121225 IBSP!OBP!SRG11!5
1416 003252 177777 -1
```

```

1417 003254
1418 003254 105077 177200
1419 003260 112777 000100 177172
1420 003266 142777 000300 177164
1421 003274 004737 002770
1422 003300 000205
1423
1424
1425
1426
1427
1428
1429
1430 003302
1431 003302 010537 003322
1432 003306 013701 002456
1433 003312 000240
1434 003314 000240
1435 003316 000240
1436 003320 000205
1437 003322 000000
1438 003324 000144
1439
1440
1441
1442 003634
1443
1444 003634
1445 003634 000240
1446 003636 000240
1447 003640 112777 000002 176612
1448 003646 012577 176614
1449 003652 152777 000003 176600
1450 003660 142777 000007 176572
1451 003666 000205
1452
1453 003670
1454 003670 000240
1455 003672 000240
1456 003674 112777 000002 176556
1457 003702 012577 176560
1458 003706 000240
1459 003710 000240
1460 003712 142777 000007 176540
1461 003720
1462 003720 152777 000001 176532
1463 003726 142777 000007 176524
1464 003734 000240
1465 003736 000240
1466 003740 152777 000002 176512
1467 003746
1468 003746 000205
1469 003750
1470
1471 003750
1472 003754 000400

```

```

.MSTCLR:
CLR B @KMCSRH ;*: CLEAR RUN, IF UP
MOV B #BIT6,@KMCSRH ;SET INST.
BIC B #BIT6!BIT7,@KMCSRH
JSR PC,CLRKM
RTS R5

:
:DEBUG THIS ROUTINE IS ENTERED AT THE BEGINING OF EACH TEST
: IN ORDER TO LOAD THE CSR ADDR. INTO R1..
: ALSO THIS PROGRAM TRACKS ITSELF HERE.
: CALL=JSR R5,DEBUG
:

DEBUG:
MOV R5,TESTAD ;SAVE ADDR. OF TEST.
MOV KMCSR,R1 ;LOAD KMCSR INTP R1.
NOP
NOP
NOP
NOP
RTS R5

TESTAD: .WORD 0 ;LAST TEST ADDR.
PATCH: .BLKW 100. ;PATCH AREA.

ENDBUG:
: UNSAFE TO PATCH ANY OTHER AREA.
.ROMCLK:
NOP
NOP
.REGT: MOV B #BIT1,@KMCSRH
MOV (R5)+,@KMPO6
BIS B #BIT1!BIT0,@KMCSRH
BIC B #BIT2!BIT1!BIT0,@KMCSRH
RTS R5

.SROMCLK:
NOP
NOP
MOV B #BIT1,@KMCSRH
MOV (R5)+,@KMPO6
NOP
NOP
BIC B #7,@KMCSRH
1$: BIS B #BIT0,@KMCSRH ;STEP INSTR.
BIC B #7,@KMCSRH
NOP
NOP
BIS B #BIT1,@KMCSRH
2$: RTS R5

CLRALL: ;CLEARS C & Z BITS AND BR
ROMCLK
400 ;0 TO BR

```



```
1473 003756 ROMCLK  
1474 003762 063220 63220 ;SP(0) TO BR  
1475 003764 ROMCLK  
1476 003770 060400 60400 ;BR,SP(0) + BR  
1477 003772 SRMCLK  
1478 003776 000000 0  
1479 004000 000207 RTS PC  
1480  
1481 004002 SETBR0:  
1482 ;SETS BR0 BIT  
1483 004002 ROMCLK  
1484 004006 000401 401 ;1 TO BR  
1485 004010 000207 RTS PC  
1486  
1487 004012 SETBR1:  
1488 ;THIS SUBROUTINE SETS BR1 BIT  
1489  
1490 004012 ROMCLK ;NEXT WORD IS INSTRUCTION  
1491 004016 000402 000402 ;BR_002  
1492 004020 000207 RTS PC  
1493  
1494 004022 SETBR4:  
1495 ;THIS SUBROUTINE SETS BR4 BIT  
1496  
1497 004022 ROMCLK ;NEXT WORD IS INSTRUCTION  
1498 004026 000420 420  
1499 004030 000207 RTS PC  
1500  
1501 004032 SETBR7:  
1502 ;THIS SUBROUTINE SETS BR7 BIT  
1503  
1504 004032 ROMCLK ;NEXT WORD IS INSTRUCTION  
1505 004036 000600 600  
1506 004040 000207 RTS PC  
1507  
1508 004042 SETC:  
1509 ;THIS SUBROUTINE SETS THE C BIT  
1510  
1511 004042 ROMCLK ;NEXT WORD IS INSTRUCTION  
1512 004046 000777 000777 ;BR 377  
1513 004050 ROMCLK ;NEXT WORD IS INSTRUCTION  
1514 004054 063220 063220 ;SP(0) BR  
1515 004056 ROMCLK ;NEXT WORD IS INSTRUCTION  
1516 004062 060400 060400 ;BR SP(0)+BR  
1517 004064 SRMCLK ;NOW WE MUST CLOCK THE BITS INTO IBUS <13>  
1518 004070 000000 0  
1519 004072 000207 RTS PC  
1520  
1521 004074 SETZ:  
1522 ;THIS SUBROUTINE SETS THE Z BIT  
1523  
1524 004074 ROMCLK ;NEXT WORD IS INSTRUCTION  
1525 004100 000777 000777 ;BR 377  
1526 004102 SRMCLK ;NOW CLOCK THE BITS INTO IBUS<13>  
1527 004106 000777 0777  
1528 004110 000207 RTS PC
```

```

1529
1530 004112      RAMDAT:
1531              ;THIS SUBROUTINE LOADS R4 WITH THE CRAM PC
1532
1533 004112 017605 000000      MOV    @ (SP),R5      ;GOOD DATA
1534 004116 062716 000002      ADD    #2,(SP)      ;ADJUST STACK
1535 004122      1$: ROMCLK      ;READ PC LOW REG DIRECTLY.
1536 004126 121244      121244      ;IBUS*<12> TO PORT 4
1537 004130      ROMCLK
1538 004134 121265      121265      ; IBUS*<13> TO PORT 5
1539 004136 016104 000004      MOV    4(R1),R4      ;PUT INTO R4
1540 004142 042704 140000      BIC   #140000,R4    ; CLEAR C AND Z
1541 004146 000207      RTS   PC            ;EXIT
1542
1543 004150      WROM:
1544              ;THIS SUBROUTINE WRITES THE ROMMAP INTO THE CRAM
1545
1546              ;
1547              ;
1548 004150      BIT    #BIT15,STAT1 ;BE SURE M8200,4,6,7 HAS CRAM
1549 004160 005000      BEQ   2$           ;SKIP IF NO CRAM
1550 004162 012702 013116      SKIP07 2$
1551 004166 012711 002000      CLR   R0           ;R0=CRAM ADDRESS
1552 004172 010061 000004      MOV   #ROMMAP,R2   ;R2 POINTS TO ROMMAP
1553 004176 012261 000006      1$: MOV   #BIT10,(R1) ;SET ROMO
1554 004202 052711 020000      MOV   R0,4(R1)     ;LOAD CRAM ADDRESS
1555 004206 005200      MOV   (R2)+,6(R1)  ;LOAD WORD TO BE WRITTEN
1556 004210 023700 002400      BIS   #BIT13,(R1)  ;WRITE IT!
1557 004214 001364      INC   R0           ;NEXT ADDRESS
1558 004216 005011      CMP   MEMSZ,R0     ;DONE YET?
1559 004220 000207      BNE   1$           ;BR IF NO
1560              CLR   (R1)      ;CLEAR SELO
1561 004222      2$: RTS   PC            ;RETURN
1562
1563      MEMSET:
1564              ;THIS SUBROUTINE LOADS CRAM WITH SPECIAL INSTRUCTIONS
1565              ;FOR THE CRAM JUMP TEST. ALL CRAM LOCATIONS ARE LOADED
1566              ;WITH INSTRUCTIONS THAT MOVE B7-0 OF ITS ADDRESS TO BRG.
1567 004222 005000      CLR   R0           ;R0 = CRAM ADDRESS
1568 004224 012737 000400 004270      MOV   #400,5$      ; LOAD BR_0
1569 004232 012711 002000      MOV   #BIT10,(R1)  ;SET ROMO
1570 004236 010061 000004      1$: MOV   R0,4(R1)     ;LOAD CRAM ADDRESS
1571 004242 013761 004270 000006      MOV   5$,6(R1)     ;LOAD INSTRUCTION
1572 004250 105237 004270      INCB  5$           ; BUMP INSTRUCTION
1573 004254 005200      INC   R0           ;NEXT ADDRESS
1574 004256 023700 002400      CMP   MEMSZ,R0     ;DONE YET?
1575 004262 101765      BLOS  1$           ;BR IF NO
1576 004264 005011      CLR   (R1)         ;CLEAR ALL BITS
1577 004266 000207      3$: RTS   PC            ;RETURN
1578 004270 000000      5$: 0
1579 004272 000000 000001 000004 CRAMA: .WORD 0,1,4,7,1777,525
1580 004300 000007 001777 000525
1581
1582 004306 000400      INSTU: 000400      ;BR_0
1583 004310 000401      000401      ;BR_1
1584 004312 000404      000404      ;BR_4

```

1585 004314 000407
1586 004316 000777
1587 004320 000525
1588
1589
1590
1591
1592 004322 010537 002360
1593 004326 010437 002362
1594 004332 010337 002364
1595 004336 010237 002366
1596 004342 010137 002370
1597 004346 013737 002376 002372
1598 004354 000207
1599
1600

000407 :BR_7
000777 :BR_377
000525 :BR_125

:ROUTINE TO SAVE GENERAL REGISTERS FOR ERROR ROUTINE.

: CALL = JSR PC,SV05

SV05:

MOV R5,\$REG5
MOV R4,\$REG4
MOV R3,\$REG3
MOV R2,\$REG2
MOV R1,\$REG1
MOV MR0,\$REG0
RTS PC

1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656

004356	047045	047445	022466
004364	032123	047445	022466
004372	032123	047445	022464
004400	000116		
004402	047045	047445	022463
004410	033523	047445	022463
004416	000116		
004420	047045	047445	022463
004426	030523	022460	031517
004434	051445	022464	032117
004442	047045	000	
004445	045	022516	033117
004452	051445	022465	033117
004460	047045	000	
004463	045	022516	033117
004470	051445	022465	033117
004476	051445	022463	033117
004504	047045	000	
004507	045	022516	051101
004514	043505	051511	042524
004522	020122	042101	051104
004530	051505	020123	051105
004536	047522	026122	042101
004544	051104	051505	020123
004552	020075	047445	022466
004560	026101	047125	052111
004566	036440	022440	031117
004574	000		
004575	045	022516	020101
004602	051503	020122	044510
004610	044107	041040	052131
004616	020105	047507	020124
004624	051127	052111	042524
004632	020116	047111	047524
004640	047440	020116	020101
004646	047514	020127	054502
004654	042524	054040	042506
004662	000122		
004664	047045	040445	041440
004672	051123	046040	053517
004700	041040	052131	020105
004706	047507	020124	051127
004714	052111	042524	020116
004722	047111	047524	047440
004730	020116	020101	044510

.SBTTL GLOBAL ERROR REPORT SECTION

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

TFM1: .ASCIZ /%N%06%S4%06%S4%04%N/

TFM2: .ASCIZ /%N%03%S7%03%N/

TFM3: .ASCIZ /%N%03%S10%03%S4%04%N/

TFM4: .ASCIZ /%N%06%S5%06%N/

TFM5: .ASCIZ /%N%06%S5%06%S3%06%N/

TFM36: .ASCIZ /%N%AREGISTER ADDRESS ERROR,ADDRESS = %06%A,UNIT = %02/

TFM41: .ASCIZ /%N%A CSR HIGH BYTE GOT WRITTEN INTO ON A LOW BYTE XFER/

TFM42: .ASCIZ /%N%A CSR LOW BYTE GOT WRITTEN INTO ON A HIGH BYTE XFER/

1657	004736	044107	041040	052131	
1658	004744	020105	043130	051105	
1659	004752	000			
1660	004753	045	022516	020101	TFM40: .ASCIZ /%N%A NEG ADDR TEST DUAL ADDR ERROR-BAD ADDR = %06/
1661	004760	042516	020107	042101	
1662	004766	051104	052040	051505	
1663	004774	020124	052504	046101	
1664	005002	040440	042104	020122	
1665	005010	051105	047522	026522	
1666	005016	040502	020104	042101	
1667	005024	051104	036440	022440	
1668	005032	033117	000		
1669	005035	045	020101	041523	TFM43: .ASCIZ /%A SCRATCH PAD %03%A DUAL ADDRESS ERROR WITH SP%02/
1670	005042	040522	041524	020110	
1671	005050	040520	020104	047445	
1672	005056	022463	020101	052504	
1673	005064	046101	040440	042104	
1674	005072	042522	051523	042440	
1675	005100	051122	051117	053440	
1676	005106	052111	020110	050123	
1677	005114	047445	000062		
1678	005120	052045	040445	044124	TFM44: .ASCIZ /%T%THE MAR REG, CONTENTS= %06/
1679	005126	020105	040515	020122	
1680	005134	042522	026107	041440	
1681	005142	047117	042524	052116	
1682	005150	036523	022440	033117	
1683	005156	000			
1684	005157	045	022524	052101	TFM45: .ASCIZ /%T%THE PC REG, CONTENTS= %06/
1685	005164	042510	050040	020103	
1686	005172	042522	026107	041440	
1687	005200	047117	042524	052116	
1688	005206	036523	022440	033117	
1689	005214	000			
1690	005215	045	047101	051120	TFM46: .ASCIZ "%ANPR/MISC REGS DATA FAILURE, GOOD =%06%A, BAD =%06"
1691	005222	046457	051511	020103	
1692	005230	042522	051507	042040	
1693	005236	052101	020101	040506	
1694	005244	046111	051125	026105	
1695	005252	043440	047517	020104	
1696	005260	022475	033117	040445	
1697	005266	020054	040502	020104	
1698	005274	022475	033117	000	
1699	005301	115	051501	042524	TMMC: .ASCIZ /MASTER CLEAR FAILED TO CLEAR /
1700	005306	020122	046103	040505	
1701	005314	020122	040506	046111	
1702	005322	042105	052040	020117	
1703	005330	046103	040505	020122	
1704	005336	000			
1705	005337	045	022516	022524	FM1: .ASCIZ /%N%T%N/
1706	005344	000116			
1707	005346	047045	040445	047125	FMX: .ASCIZ /%N%AUNIT= %02%A ; FAILING UNIT ADDRESS = %06%N/
1708	005354	052111	020075	047445	
1709	005362	022462	020101	020073	
1710	005370	040506	046111	047111	
1711	005376	020107	047125	052111	
1712	005404	040440	042104	042522	

1713	005412	051523	036440	022440	
1714	005420	033117	047045	000	
1715					
1716					
1717					
1718	005425	000			EM0: .ASCIZ //
1719	005426	051103	046501	042040	EM1: .ASCIZ /CRAM DATA ERROR/
1720	005434	052101	020101	051105	
1721	005442	047522	000122		
1722	005446	051103	046501	042040	EM2: .ASCIZ /CRAM DUAL ADDRESSING ERROR/
1723	005454	040525	020114	042101	
1724	005462	051104	051505	044523	
1725	005470	043516	042440	051122	
1726	005476	051117	000		
1727	005501	112	046525	020120	EM3: .ASCIZ /JUMP ERROR/
1728	005506	051105	047522	000122	
1729	005514	051103	046501	045040	EM4: .ASCIZ /CRAM JUMP TEST FAULT/
1730	005522	046525	020120	042524	
1731	005530	052123	043040	052501	
1732	005536	052114	000		
1733	005541	111	050117	046440	EM5: .ASCIZ /IOP MAIN MEMORY TEST/
1734	005546	044501	020116	042515	
1735	005554	047515	054522	052040	
1736	005562	051505	000124		
1737	005566	047511	020120	040515	EM6: .ASCIZ /IOP MAR TEST/
1738	005574	020122	042524	052123	
1739	005602	000			
1740	005603	102	020122	044522	EM7: .ASCIZ /BR RIGHT SHIFT ERROR/
1741	005610	044107	020124	044123	
1742	005616	043111	020124	051105	
1743	005624	047522	000122		
1744	005630	040515	020122	052504	EM10: .ASCIZ /MAR DUAL ADDRESSING ERROR/
1745	005636	046101	040440	042104	
1746	005644	042522	051523	047111	
1747	005652	020107	051105	047522	
1748	005660	000122			
1749	005662	052512	050115	043040	EM11: .ASCIZ /JUMP FIELD ERROR/
1750	005670	042511	042114	042440	
1751	005676	051122	051117	000	
1752	005703	112	046525	020120	EM12: .ASCIZ /JUMP TEST ERROR/
1753	005710	042524	052123	042440	
1754	005716	051122	051117	000	
1755	005723	103	047117	044504	EM16: .ASCIZ /CONDITION CODE TESTING,Z & C/
1756	005730	044524	047117	041440	
1757	005736	042117	020105	042524	
1758	005744	052123	047111	026107	
1759	005752	020132	020046	000103	
1760	005760				EM35:
1761	005760	047506	041522	020105	EM17: .ASCIZ /FORCE POWER FAIL ERROR/
1762	005766	047520	042527	020122	
1763	005774	040506	046111	042440	
1764	006002	051122	051117	000	
1765	006007	111	052502	025123	EM27: .ASCIZ "IBUS* WRITE/READ ERROR"
1766	006014	053440	044522	042524	
1767	006022	051057	040505	020104	
1768	006030	051105	047522	000122	


```
1769
1770 006036 041111 051525 047457 EM29: .ASCIZ 'IBUS/OBUS WRITE/READ ERROR'
1771 006044 052502 020123 051127
1772 006052 052111 027505 042522
1773 006060 042101 042440 051122
1774 006066 051117 000
1775 006071 045 022516 025101 STM: .ASCIZ '%N%A*****'
1776 006076 025052 025052 025052
1777 006104 025052 025052 025052
1778 006112 025052 025052 025052
1779 006120 025052 025052 025052
1780 006126 025052 025052 025052
1781 006134 025052 025052 025052
1782 006142 025052 025052 025052
1783 006150 025052 025052 025052
1784 006156 025052 025052 025052
1785 006164 000
1786 006165 000 DH0: .ASCIZ //
1787
1788
1789 006166 054105 042520 052103 DH1: .ASCIZ /EXPECTED FOUND ADDRESS/
1790 006174 042105 020040 047506
1791 006202 047125 020104 040440
1792 006210 042104 042522 051523
1793 006216 000
1794 006217 105 050130 041505 DH2: .ASCIZ /EXPECTED FOUND/
1795 006224 042524 020104 043040
1796 006232 052517 042116 000
1797 006237 106 047522 020115 DH3: .ASCIZ /FROM ADDR TO ADDR BAD ADDR/
1798 006244 042101 051104 020040
1799 006252 047524 040440 042104
1800 006260 020122 041040 042101
1801 006266 040440 042104 000122
1802
1803
1804 .EVEN
1805
1806
1807
1808 -----
1809 : MACRO'S NEEDED TO REPORT ERRORS
1810 : -----
1811
1812 .MACRO MDT1
1813 PRINTB #TFM1,$REG2,$REG4,$REG0
1814 .ENDM
1815
1816 .MACRO MDT2
1817 PRINTB #TFM1,$REG5,$REG4,$REG2
1818 .ENDM
1819
1820 .MACRO MDT3
1821 PRINTB #TFM2,$REG5,$REG4
1822 .ENDM
1823
1824 .MACRO MDT4
```

```
1825          PRINTB #TFM3,$REG5,$REG4,FLAG
1826      .ENDM
1827
1828      .MACRO MDT5
1829          PRINTB #TFM3,$REG5,$REG4,$REG2
1830      .ENDM
1831
1832      .MACRO MDT0
1833      .ENDM
1834      .MACRO MDT6
1835          PRINTB #TFM4,$REG2,$REG4
1836      .ENDM
1837
1838      .MACRO MDT7
1839          PRINTB #TFM4,$REG5,$REG4
```

CZKMAO KMC11-B STATIC PART2
CZKMA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 ^{G 4} PAGE 45
GLOBAL ERROR REPORT SECTION

SEQ 0045

1840
1841
1842
1843

.ENDM
.MACRO MDT8
PRINTB #TFM5,FADR,\$REG5,\$REG4
.ENDM


```
1844  
1845 .MACRO SMD ERRNN ERNB ERHM ERFM  
1846 BGNMSG ERR'ERRNN  
1847 PRINTB #FMX,LOGDEV,KMCSR  
1848 PRINTB #FM1,#EM'ERNB  
1849 PRINTB #FM1,#DH'ERHM  
1850 MDT'ERFM  
1851 PRINTB #STM  
1852 ENDMSG  
1853 .ENDM  
1854 .MACRO ERROR ECB  
1855 JSR PC,SV05  
1856 ERRDF 'ECB',EMO,ERR'ECB'  
1857 .ENDM  
1858
```

1859	006274				
1860	006274			ERR1::	SMD 1,1,1,1
1861	006274	013746	002456		MOV KMCSR,-(SP)
1862	006300	013746	002304		MOV LOGDEV,-(SP)
1863	006304	012746	005346		MOV #FMX,-(SP)
1864	006310	012746	000003		MOV #3,-(SP)
1865	006314	010600			MOV SP,R0
1866	006316	104414			TRAP C\$PNTB
1867	006320	062706	000010		ADD #10,SP
1868	006324	012746	005426		MOV #EM1,-(SP)
1869	006330	012746	005337		MOV #FM1,-(SP)
1870	006334	012746	000002		MOV #2,-(SP)
1871	006340	010600			MOV SP,R0
1872	006342	104414			TRAP C\$PNTB
1873	006344	062706	000006		ADD #6,SP
1874	006350	012746	006166		MOV #DH1,-(SP)
1875	006354	012746	005337		MOV #FM1,-(SP)
1876	006360	012746	000002		MOV #2,-(SP)
1877	006364	010600			MOV SP,R0
1878	006366	104414			TRAP C\$PNTB
1879	006370	062706	000006		ADD #6,SP
1880	006374	013746	002372		MOV \$REG0,-(SP)
1881	006400	013746	002362		MOV \$REG4,-(SP)
1882	006404	013746	002366		MOV \$REG2,-(SP)
1883	006410	012746	004356		MOV #TFM1,-(SP)
1884	006414	012746	000004		MOV #4,-(SP)
1885	006420	010600			MOV SP,R0
1886	006422	104414			TRAP C\$PNTB
1887	006424	062706	000012		ADD #12,SP
1888	006430	012746	006071		MOV #STM,-(SP)
1889	006434	012746	000001		MOV #1,-(SP)
1890	006440	010600			MOV SP,R0
1891	006442	104414			TRAP C\$PNTB
1892	006444	062706	000004		ADD #4,SP
1893	006450			L10002:	TRAP C\$MSG
1894	006450	104423			SMD 2,2,1,1
1895	006452			ERR2::	
1896	006452				MOV KMCSR,-(SP)
1897	006452	013746	002456		MOV LOGDEV,-(SP)
1898	006456	013746	002304		MOV #FMX,-(SP)
1899	006462	012746	005346		MOV #3,-(SP)
1900	006466	012746	000003		MOV SP,R0
1901	006472	010600			TRAP C\$PNTB
1902	006474	104414			ADD #10,SP
1903	006476	062706	000010		MOV #EM2,-(SP)
1904	006502	012746	005446		MOV #FM1,-(SP)
1905	006506	012746	005337		MOV #2,-(SP)
1906	006512	012746	000002		MOV SP,R0
1907	006516	010600			TRAP C\$PNTB
1908	006520	104414			ADD #6,SP
1909	006522	062706	000006		MOV #DH1,-(SP)
1910	006526	012746	006166		MOV #FM1,-(SP)
1911	006532	012746	005337		MOV #2,-(SP)
1912	006536	012746	000002		MOV SP,R0
1913	006542	010600			TRAP C\$PNTB
1914	006544	104414			

1915	006546	062706	000006	ADD	#6,SP
1916	006552	013746	002372	MOV	\$REG0,-(SP)
1917	006556	013746	002362	MOV	\$REG4,-(SP)
1918	006562	013746	002366	MOV	\$REG2,-(SP)
1919	006566	012746	004356	MOV	#TFM1,-(SP)
1920	006572	012746	000004	MOV	#4,-(SP)
1921	006576	010600		MOV	SP,R0
1922	006600	104414		TRAP	C\$PNTB
1923	006602	062706	000012	ADD	#12,SP
1924	006606	012746	006071	MOV	#STM,-(SP)
1925	006612	012746	000001	MOV	#1,-(SP)
1926	006616	010600		MOV	SP,R0
1927	006620	104414		TRAP	C\$PNTB
1928	006622	062706	000004	ADD	#4,SP
1929	006626				
1930	006626	104423		L10003: TRAP	C\$MSG
1931	006630			\$MD	3,1,1,2
1932	006630			ERR3::	
1933	006630	013746	002456	MOV	KMCSR,-(SP)
1934	006634	013746	002304	MOV	LOGDEV,-(SP)
1935	006640	012746	005346	MOV	#FMX,-(SP)
1936	006644	012746	000003	MOV	#3,-(SP)
1937	006650	010600		MOV	SP,R0
1938	006652	104414		TRAP	C\$PNTB
1939	006654	062706	000010	ADD	#10,SP
1940	006660	012746	005426	MOV	#EM1,-(SP)
1941	006664	012746	005337	MOV	#FM1,-(SP)
1942	006670	012746	000002	MOV	#2,-(SP)
1943	006674	010600		MOV	SP,R0
1944	006676	104414		TRAP	C\$PNTB
1945	006700	062706	000006	ADD	#6,SP
1946	006704	012746	006166	MOV	#DH1,-(SP)
1947	006710	012746	005337	MOV	#FM1,-(SP)
1948	006714	012746	000002	MOV	#2,-(SP)
1949	006720	010600		MOV	SP,R0
1950	006722	104414		TRAP	C\$PNTB
1951	006724	062706	000006	ADD	#6,SP
1952	006730	013746	002366	MOV	\$REG2,-(SP)
1953	006734	013746	002362	MOV	\$REG4,-(SP)
1954	006740	013746	002360	MOV	\$REG5,-(SP)
1955	006744	012746	004356	MOV	#TFM1,-(SP)
1956	006750	012746	000004	MOV	#4,-(SP)
1957	006754	010600		MOV	SP,R0
1958	006756	104414		TRAP	C\$PNTB
1959	006760	062706	000012	ADD	#12,SP
1960	006764	012746	006071	MOV	#STM,-(SP)
1961	006770	012746	000001	MOV	#1,-(SP)
1962	006774	010600		MOV	SP,R0
1963	006776	104414		TRAP	C\$PNTB
1964	007000	062706	000004	ADD	#4,SP
1965	007004			L10004: TRAP	C\$MSG
1966	007004	104423		\$MD	4,3,2,3
1967	007006			ERR4::	
1968	007006				
1969	007006	013746	002456	MOV	KMCSR,-(SP)
1970	007012	013746	002304	MOV	LOGDEV,-(SP)

1971	007016	012746	005346	MOV	#FMX,-(SP)
1972	007022	012746	000003	MOV	#3,-(SP)
1973	007026	010600		MOV	SP,R0
1974	007030	104414		TRAP	C\$PNTB
1975	007032	062706	000010	ADD	#10,SP
1976	007036	012746	005501	MOV	#EM3,-(SP)
1977	007042	012746	005337	MOV	#FM1,-(SP)
1978	007046	012746	000002	MOV	#2,-(SP)
1979	007052	010600		MOV	SP,R0
1980	007054	104414		TRAP	C\$PNTB
1981	007056	062706	000006	ADD	#6,SP
1982	007062	012746	006217	MOV	#DH2,-(SP)
1983	007066	012746	005337	MOV	#FM1,-(SP)
1984	007072	012746	000002	MOV	#2,-(SP)
1985	007076	010600		MOV	SP,R0
1986	007100	104414		TRAP	C\$PNTB
1987	007102	062706	000006	ADD	#6,SP
1988	007106	013746	002362	MOV	\$REG4,-(SP)
1989	007112	013746	002360	MOV	\$REG5,-(SP)
1990	007116	012746	004402	MOV	#TFM2,-(SP)
1991	007122	012746	000003	MOV	#3,-(SP)
1992	007126	010600		MOV	SP,R0
1993	007130	104414		TRAP	C\$PNTB
1994	007132	062706	000010	ADD	#10,SP
1995	007136	012746	006071	MOV	#STM,-(SP)
1996	007142	012746	000001	MOV	#1,-(SP)
1997	007146	010600		MOV	SP,R0
1998	007150	104414		TRAP	C\$PNTB
1999	007152	062706	000004	ADD	#4,SP
2000	007156				
2001	007156	104423		L10005: TRAP	C\$MSG

Address	Instruction	Source	Destination
2002		007160	
2003		007160	
2004		007160	013746 002456
2005		007164	013746 002304
2006		007170	012746 005346
2007		007174	012746 000003
2008		007200	010600
2009		007202	104414
2010		007204	062706 000010
2011		007210	012746 005514
2012		007214	012746 005337
2013		007220	012746 000002
2014		007224	010600
2015		007226	104414
2016		007230	062706 000006
2017		007234	012746 006217
2018		007240	012746 005337
2019		007244	012746 000002
2020		007250	010600
2021		007252	104414
2022		007254	062706 000006
2023		007260	013746 002362
2024		007264	013746 002360
2025		007270	012746 004402
2026		007274	012746 000003
2027		007300	010600
2028		007302	104414
2029		007304	062706 000010
2030		007310	012746 006071
2031		007314	012746 000001
2032		007320	010600
2033		007322	104414
2034		007324	062706 000004
2035		007330	
2036		007330	104423

ERR5:: SMD 5,4,2,3

MOV KMCSR,-(SP)
MOV LOGDEV,-(SP)
MOV #FMX,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #10,SP
MOV #EM4,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #6,SP
MOV #DH2,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #6,SP
MOV \$REG4,-(SP)
MOV \$REG5,-(SP)
MOV #TFM2,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #10,SP
MOV #STM,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #4,SP

L10006: TRAP C\$MSG

2037 007332
2038 007332
2039 007332 013746 002456
2040 007336 013746 002304
2041 007342 012746 005346
2042 007346 012746 000003
2043 007352 010600
2044 007354 104414
2045 007356 062706 000010
2046 007362 012746 005541
2047 007366 012746 005337
2048 007372 012746 000002
2049 007376 010600
2050 007400 104414
2051 007402 062706 000006
2052 007406 012746 006166
2053 007412 012746 005337
2054 007416 012746 000002
2055 007422 010600
2056 007424 104414
2057 007426 062706 000006
2058 007432 013746 002350
2059 007436 013746 002362
2060 007442 013746 002360
2061 007446 012746 004420
2062 007452 012746 000004
2063 007456 010600
2064 007460 104414
2065 007462 062706 000012
2066 007466 012746 006071
2067 007472 012746 000001
2068 007476 010600
2069 007500 104414
2070 007502 062706 000004
2071 007506
2072 007506 104423
2073 007510
2074 007510
2075 007510 013746 002456
2076 007514 013746 002304
2077 007520 012746 005346
2078 007524 012746 000003
2079 007530 010600
2080 007532 104414
2081 007534 062706 000010
2082 007540 012746 005566
2083 007544 012746 005337
2084 007550 012746 000002
2085 007554 010600
2086 007556 104414
2087 007560 062706 000006
2088 007564 012746 006166
2089 007570 012746 005337
2090 007574 012746 000002
2091 007600 010600
2092 007602 104414

ERR6:: SMD 6.5.1.4
MOV KMCSR,-(SP)
MOV LOGDEV,-(SP)
MOV #FMX,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #10,SP
MOV #EM5,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #6,SP
MOV #DH1,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #6,SP
MOV FLAG,-(SP)
MOV \$REG4,-(SP)
MOV \$REG5,-(SP)
MOV #TFM3,-(SP)
MOV #4,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #12,SP
MOV #STM,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #4,SP
L10007: TRAP C\$MSG
ERR7:: SMD 7.6.1.5
MOV KMCSR,-(SP)
MOV LOGDEV,-(SP)
MOV #FMX,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #10,SP
MOV #EM6,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #6,SP
MOV #DH1,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTB

2093	007604	062706	000006	ADD	#6,SP
2094	007610	013746	002366	MOV	\$REG2,-(SP)
2095	007614	013746	002362	MOV	\$REG4,-(SP)
2096	007620	013746	002360	MOV	\$REG5,-(SP)
2097	007624	012746	004420	MOV	#TFM3,-(SP)
2098	007630	012746	000004	MOV	#4,-(SP)
2099	007634	010600		MOV	SP,R0
2100	007636	104414		TRAP	CSPNTB
2101	007640	062706	000012	ADD	#12,SP
2102	007644	012746	006071	MOV	#STM,-(SP)
2103	007650	012746	000001	MOV	#1,-(SP)
2104	007654	010600		MOV	SP,R0
2105	007656	104414		TRAP	CSPNTB
2106	007660	062706	000004	ADD	#4,SP
2107	007664				
2108	007664	104423		L10010: TRAP	C\$MSG
2109	007666			SMD	10.7.2.3
2110	007666			ERR10::	
2111	007666	013746	002456	MOV	KMCSR,-(SP)
2112	007672	013746	002304	MOV	LOGDEV,-(SP)
2113	007676	012746	005346	MOV	#FMX,-(SP)
2114	007702	012746	000003	MOV	#3,-(SP)
2115	007706	010600		MOV	SP,R0
2116	007710	104414		TRAP	CSPNTB
2117	007712	062706	000010	ADD	#10,SP
2118	007716	012746	005603	MOV	#EM7,-(SP)
2119	007722	012746	005337	MOV	#FM1,-(SP)
2120	007726	012746	000002	MOV	#2,-(SP)
2121	007732	010600		MOV	SP,R0
2122	007734	104414		TRAP	CSPNTB
2123	007736	062706	000006	ADD	#6,SP
2124	007742	012746	006217	MOV	#DH2,-(SP)
2125	007746	012746	005337	MOV	#FM1,-(SP)
2126	007752	012746	000002	MOV	#2,-(SP)
2127	007756	010600		MOV	SP,R0
2128	007760	104414		TRAP	CSPNTB
2129	007762	062706	000006	ADD	#6,SP
2130	007766	013746	002362	MOV	\$REG4,-(SP)
2131	007772	013746	002360	MOV	\$REG5,-(SP)
2132	007776	012746	004402	MOV	#TFM2,-(SP)
2133	010002	012746	000003	MOV	#3,-(SP)
2134	010006	010600		MOV	SP,R0
2135	010010	104414		TRAP	CSPNTB
2136	010012	062706	000010	ADD	#10,SP
2137	010016	012746	006071	MOV	#STM,-(SP)
2138	010022	012746	000001	MOV	#1,-(SP)
2139	010026	010600		MOV	SP,R0
2140	010030	104414		TRAP	CSPNTB
2141	010032	062706	000004	ADD	#4,SP
2142	010036			L10011: TRAP	C\$MSG
2143	010036	104423		SMD	11.10.2.6
2144	010040			ERR11::	
2145	010040				
2146	010040	013746	002456	MOV	KMCSR,-(SP)
2147	010044	013746	002304	MOV	LOGDEV,-(SP)
2148	010050	012746	005346	MOV	#FMX,-(SP)

2149	010054	012746	000003	MOV	#3,-(SP)
2150	010060	010600		MOV	SP,R0
2151	010062	104414		TRAP	C\$PNTB
2152	010064	062706	000010	ADD	#10,SP
2153	010070	012746	005630	MOV	#EM10,-(SP)
2154	010074	012746	005337	MOV	#FM1,-(SP)
2155	010100	012746	000002	MOV	#2,-(SP)
2156	010104	010600		MOV	SP,R0
2157	010106	104414		TRAP	C\$PNTB
2158	010110	062706	000006	ADD	#6,SP
2159	010114	012746	006217	MOV	#DH2,-(SP)
2160	010120	012746	005337	MOV	#FM1,-(SP)
2161	010124	012746	000002	MOV	#2,-(SP)
2162	010130	010600		MOV	SP,R0
2163	010132	104414		TRAP	C\$PNTB
2164	010134	062706	000006	ADD	#6,SP
2165	010140	013746	002362	MOV	\$REG4,-(SP)
2166	010144	013746	002366	MOV	\$REG2,-(SP)
2167	010150	012746	004445	MOV	#TFM4,-(SP)
2168	010154	012746	000003	MOV	#3,-(SP)
2169	010160	010600		MOV	SP,R0
2170	010162	104414		TRAP	C\$PNTB
2171	010164	062706	000010	ADD	#10,SP
2172	010170	012746	006071	MOV	#STM,-(SP)
2173	010174	012746	000001	MOV	#1,-(SP)
2174	010200	010600		MOV	SP,R0
2175	010202	104414		TRAP	C\$PNTB
2176	010204	062706	000004	ADD	#4,SP
2177	010210				
2178	010210	104423		L10012: TRAP	C\$MSG
2179	010212			\$MD	12,7,2,7
2180	010212			ERR12::	
2181	010212	013746	002456	MOV	KMCSR,-(SP)
2182	010216	013746	002304	MOV	LOGDEV,-(SP)
2183	010222	012746	005346	MOV	#FMX,-(SP)
2184	010226	012746	000003	MOV	#3,-(SP)
2185	010232	010600		MOV	SP,R0
2186	010234	104414		TRAP	C\$PNTB
2187	010236	062706	000010	ADD	#10,SP
2188	010242	012746	005603	MOV	#EM7,-(SP)
2189	010246	012746	005337	MOV	#FM1,-(SP)
2190	010252	012746	000002	MOV	#2,-(SP)
2191	010256	010600		MOV	SP,R0
2192	010260	104414		TRAP	C\$PNTB
2193	010262	062706	000006	ADD	#6,SP
2194	010266	012746	006217	MOV	#DH2,-(SP)
2195	010272	012746	005337	MOV	#FM1,-(SP)
2196	010276	012746	000002	MOV	#2,-(SP)
2197	010302	010600		MOV	SP,R0
2198	010304	104414		TRAP	C\$PNTB
2199	010306	062706	000006	ADD	#6,SP
2200	010312	013746	002362	MOV	\$REG4,-(SP)
2201	010316	013746	002360	MOV	\$REG5,-(SP)
2202	010322	012746	004445	MOV	#TFM4,-(SP)
2203	010326	012746	000003	MOV	#3,-(SP)
2204	010332	010600		MOV	SP,R0

Address	Instruction	Comment
2205	TRAP	C\$PNTB
2206	ADD	#10,SP
2207	MOV	#STM,-(SP)
2208	MOV	#1,-(SP)
2209	MOV	SP,R0
2210	TRAP	C\$PNTB
2211	ADD	#4,SP
2212		
2213	TRAP	C\$MSG
2214	\$MD	13,10,2,3
2215		
2216	MOV	KMCSR,-(SP)
2217	MOV	LOGDEV,-(SP)
2218	MOV	#FMX,-(SP)
2219	MOV	#3,-(SP)
2220	MOV	SP,R0
2221	TRAP	C\$PNTB
2222	ADD	#10,SP
2223	MOV	#EM10,-(SP)
2224	MOV	#FM1,-(SP)
2225	MOV	#2,-(SP)
2226	MOV	SP,R0
2227	TRAP	C\$PNTB
2228	ADD	#6,SP
2229	MOV	#DH2,-(SP)
2230	MOV	#FM1,-(SP)
2231	MOV	#2,-(SP)
2232	MOV	SP,R0
2233	TRAP	C\$PNTB
2234	ADD	#6,SP
2235	MOV	\$REG4,-(SP)
2236	MOV	\$REG5,-(SP)
2237	MOV	#TFM2,-(SP)
2238	MOV	#3,-(SP)
2239	MOV	SP,R0
2240	TRAP	C\$PNTB
2241	ADD	#10,SP
2242	MOV	#STM,-(SP)
2243	MOV	#1,-(SP)
2244	MOV	SP,R0
2245	TRAP	C\$PNTB
2246	ADD	#4,SP
2247		
2248	TRAP	C\$MSG
2249	\$MD	14,11,2,6
2250		
2251	MOV	KMCSR,-(SP)
2252	MOV	LOGDEV,-(SP)
2253	MOV	#FMX,-(SP)
2254	MOV	#3,-(SP)
2255	MOV	SP,R0
2256	TRAP	C\$PNTB
2257	ADD	#10,SP
2258	MOV	#EM11,-(SP)
2259	MOV	#FM1,-(SP)
2260	MOV	#2,-(SP)

L10013:

ERR13::

L10014:

ERR14::

2261	010602	010600		MOV	SP,R0
2262	010604	104414		TRAP	C\$PNTB
2263	010606	062706	000006	ADD	#6,SP
2264	010612	012746	006217	MOV	#DH2,-(SP)
2265	010616	012746	005337	MOV	#FM1,-(SP)
2266	010622	012746	000002	MOV	#2,-(SP)
2267	010626	010600		MOV	SP,R0
2268	010630	104414		TRAP	C\$PNTB
2269	010632	062706	000006	ADD	#6,SP
2270	010636	013746	002362	MOV	\$REG4,-(SP)
2271	010642	013746	002366	MOV	\$REG2,-(SP)
2272	010646	012746	004445	MOV	#TFM4,-(SP)
2273	010652	012746	000003	MOV	#3,-(SP)
2274	010656	010600		MOV	SP,R0
2275	010660	104414		TRAP	C\$PNTB
2276	010662	062706	000010	ADD	#10,SP
2277	010666	012746	006071	MOV	#STM,-(SP)
2278	010672	012746	000001	MOV	#1,-(SP)
2279	010676	010600		MOV	SP,R0
2280	010700	104414		TRAP	C\$PNTB
2281	010702	062706	000004	ADD	#4,SP
2282	010706				
2283	010706	104423		L10015: TRAP	C\$MSG

2284 010710
2285 010710
2286 010710 013746 002456
2287 010714 013746 002304
2288 010720 012746 005346
2289 010724 012746 000003
2290 010730 010600
2291 010732 104414
2292 010734 062706 000010
2293 010740 012746 005703
2294 010744 012746 005337
2295 010750 012746 000002
2296 010754 010600
2297 010756 104414
2298 010760 062706 000006
2299 010764 012746 006237
2300 010770 012746 005337
2301 010774 012746 000002
2302 011000 010600
2303 011002 104414
2304 011004 062706 000006
2305 011010 013746 002362
2306 011014 013746 002360
2307 011020 013746 002354
2308 011024 012746 004463
2309 011030 012746 000004
2310 011034 010600
2311 011036 104414
2312 011040 062706 000012
2313 011044 012746 006071
2314 011050 012746 000001
2315 011054 010600
2316 011056 104414
2317 011060 062706 000004
2318 011064
2319 011064 104423
2320 011066
2321 011066
2322 011066 013746 002456
2323 011072 013746 002304
2324 011076 012746 005346
2325 011102 012746 000003
2326 011106 010600
2327 011110 104414
2328 011112 062706 000010
2329 011116 012746 005723
2330 011122 012746 005337
2331 011126 012746 000002
2332 011132 010600
2333 011134 104414
2334 011136 062706 000006
2335 011142 012746 006217
2336 011146 012746 005337
2337 011152 012746 000002
2338 011156 010600
2339 011160 104414

ERR15:: SMD 15,12,3,8
MOV KMCSR,-(SP)
MOV LOGDEV,-(SP)
MOV #FMX,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #10,SP
MOV #EM12,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #6,SP
MOV #DH3,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #6,SP
MOV \$REG4,-(SP)
MOV \$REG5,-(SP)
MOV FADR,-(SP)
MOV #TFM5,-(SP)
MOV #4,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #12,SP
MOV #STM,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #4,SP
L10016: TRAP C\$MSG
ERR16:: SMD 16,16,2,7
MOV KMCSR,-(SP)
MOV LOGDEV,-(SP)
MOV #FMX,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #10,SP
MOV #EM16,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #6,SP
MOV #DH2,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTB

2340	011162	062706	000006	ADD	#6,SP
2341	011166	013746	002362	MOV	\$REG4,-(SP)
2342	011172	013746	002360	MOV	\$REG5,-(SP)
2343	011176	012746	004445	MOV	#TFM4,-(SP)
2344	011202	012746	000003	MOV	#3,-(SP)
2345	011206	010600		MOV	SP,R0
2346	011210	104414		TRAP	C\$PNTB
2347	011212	062706	000010	ADD	#10,SP
2348	011216	012746	006071	MOV	#STM,-(SP)
2349	011222	012746	000001	MOV	#1,-(SP)
2350	011226	010600		MOV	SP,R0
2351	011230	104414		TRAP	C\$PNTB
2352	011232	062706	000004	ADD	#4,SP
2353	011236				
2354	011236	104423		TRAP	C\$MSG
2355					

L10017:

Address	Offset	Instruction	Comment
2356			
2357	011240		
2358	011240		
2359	011240	013746	002456
2360	011244	013746	002304
2361	011250	012746	005346
2362	011254	012746	000003
2363	011260	010600	
2364	011262	104414	
2365	011264	062706	000010
2366	011270	012746	005760
2367	011274	012746	005337
2368	011300	012746	000002
2369	011304	010600	
2370	011306	104414	
2371	011310	062706	000006
2372	011314	012746	006165
2373	011320	012746	005337
2374	011324	012746	000002
2375	011330	010600	
2376	011332	104414	
2377	011334	062706	000006
2378	011340	012746	006071
2379	011344	012746	000001
2380	011350	010600	
2381	011352	104414	
2382	011354	062706	000004
2383	011360		
2384	011360	104423	
2385	011362		
2386	011362		
2387	011362	013746	002456
2388	011366	013746	002304
2389	011372	012746	005346
2390	011376	012746	000003
2391	011402	010600	
2392	011404	104414	
2393	011406	062706	000010
2394	011412	012746	006036
2395	011416	012746	005337
2396	011422	012746	000002
2397	011426	010600	
2398	011430	104414	
2399	011432	062706	000006
2400	011436	012746	006217
2401	011442	012746	005337
2402	011446	012746	000002
2403	011452	010600	
2404	011454	104414	
2405	011456	062706	000006
2406	011462	013746	002362
2407	011466	013746	002360
2408	011472	012746	004402
2409	011476	012746	000003
2410	011502	010600	
2411	011504	104414	

ERR17:: SMD 17,17,0,0
MOV KMCSR,-(SP)
MOV LOGDEV,-(SP)
MOV #FMX,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #10,SP
MOV #EM17,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #6,SP
MOV #DH0,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #6,SP
MOV #STM,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #4,SP

L10020: TRAP C\$MSG
SMD 29,29,2,3

ERR29:: MOV KMCSR,-(SP)
MOV LOGDEV,-(SP)
MOV #FMX,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #10,SP
MOV #EM29,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #6,SP
MOV #DH2,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #6,SP
MOV \$REG4,-(SP)
MOV \$REG5,-(SP)
MOV #TFM2,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTB

2412	011506	062706	000010	ADD	#10,SP
2413	011512	012746	006071	MOV	#STM,-(SP)
2414	011516	012746	000001	MOV	#1,-(SP)
2415	011522	010600		MOV	SP,R0
2416	011524	104414		TRAP	CSPNTB
2417	011526	062706	000004	ADD	#4,SP
2418	011532			L10021:	
2419	011532	104423		TRAP	C\$MSG
2420	011534			\$MD	35,35,2,3
2421	011534			ERR35::	
2422	011534	013746	002456	MOV	KMCSR,-(SP)
2423	011540	013746	002304	MOV	LOGDEV,-(SP)
2424	011544	012746	005346	MOV	#FMX,-(SP)
2425	011550	012746	000003	MOV	#3,-(SP)
2426	011554	010600		MOV	SP,R0
2427	011556	104414		TRAP	CSPNTB
2428	011560	062706	000010	ADD	#10,SP
2429	011564	012746	005760	MOV	#EM35,-(SP)
2430	011570	012746	005337	MOV	#FM1,-(SP)
2431	011574	012746	000002	MOV	#2,-(SP)
2432	011600	010600		MOV	SP,R0
2433	011602	104414		TRAP	CSPNTB
2434	011604	062706	000006	ADD	#6,SP
2435	011610	012746	006217	MOV	#DH2,-(SP)
2436	011614	012746	005337	MOV	#FM1,-(SP)
2437	011620	012746	000002	MOV	#2,-(SP)
2438	011624	010600		MOV	SP,R0
2439	011626	104414		TRAP	CSPNTB
2440	011630	062706	000006	ADD	#6,SP
2441	011634	013746	002362	MOV	\$REG4,-(SP)
2442	011640	013746	002360	MOV	\$REG5,-(SP)
2443	011644	012746	004402	MOV	#TFM2,-(SP)
2444	011650	012746	000003	MOV	#3,-(SP)
2445	011654	010600		MOV	SP,R0
2446	011656	104414		TRAP	CSPNTB
2447	011660	062706	000010	ADD	#10,SP
2448	011664	012746	006071	MOV	#STM,-(SP)
2449	011670	012746	000001	MOV	#1,-(SP)
2450	011674	010600		MOV	SP,R0
2451	011676	104414		TRAP	CSPNTB
2452	011700	062706	000004	ADD	#4,SP
2453	011704			L10022:	
2454	011704	104423		TRAP	C\$MSG
2455					
2456	011706			BGNMSG	ERR36
2457	011706			ERR36::	
2458	011706			PRINTF	#TFM36,KMCSR,LOGDEV
2459	011706	013746	002304	MOV	LOGDEV,-(SP)
2460	011712	013746	002456	MOV	KMCSR,-(SP)
2461	011716	012746	004507	MOV	#TFM36,-(SP)
2462	011722	012746	000003	MOV	#3,-(SP)
2463	011726	010600		MOV	SP,R0
2464	011730	104417		TRAP	CSPNTF
2465	011732	062706	000010	ADD	#10,SP
2466	011736			PRINTB	#STM
2467	011736	012746	006071	MOV	#STM,-(SP)

2468	011742	012746	000001		MOV	#1,-(SP)
2469	011746	010600			MOV	SP,R0
2470	011750	104414			TRAP	C\$PNTB
2471	011752	062706	000004		ADD	#4,SP
2472	011756			ENDMSG		
2473	011756			L10023:		
2474	011756	104423			TRAP	C\$MSG
2475						
2476	011760			BGNMSG	ERR40	
2477	011760			ERR40::		
2478	011760			PRINTF	#TFM40,R2	
2479	011760	010246			MOV	R2,-(SP)
2480	011762	012746	004753		MOV	#TFM40,-(SP)
2481	011766	012746	000002		MOV	#2,-(SP)
2482	011772	010600			MOV	SP,R0
2483	011774	104417			TRAP	C\$PNTF
2484	011776	062706	000006		ADD	#6,SP
2485	012002			PRINTB	#STM	
2486	012002	012746	006071		MOV	#STM,-(SP)
2487	012006	012746	000001		MOV	#1,-(SP)
2488	012012	010600			MOV	SP,R0
2489	012014	104414			TRAP	C\$PNTB
2490	012016	062706	000004		ADD	#4,SP
2491	012022			ENDMSG		
2492	012022			L10024:		
2493	012022	104423			TRAP	C\$MSG
2494	012024					
2495	012024			BGNMSG	ERR41	
2496	012024			ERR41::		
2497	012024	012746	004575	PRINTF	#TFM41	
2498	012030	012746	000001		MOV	#TFM41,-(SP)
2499	012034	010600			MOV	#1,-(SP)
2500	012036	104417			MOV	SP,R0
2501	012040	062706	000004		TRAP	C\$PNTF
2502	012044				ADD	#4,SP
2503	012044	012746	006071	PRINTB	#STM	
2504	012050	012746	000001		MOV	#STM,-(SP)
2505	012054	010600			MOV	#1,-(SP)
2506	012056	104414			MOV	SP,R0
2507	012060	062706	000004		TRAP	C\$PNTB
2508	012064				ADD	#4,SP
2509	012064			ENDMSG		
2510	012064	104423		L10025:		
2511	012066				TRAP	C\$MSG
2512	012066			BGNMSG	ERR42	
2513	012066			ERR42::		
2514	012066	012746	004664	PRINTF	#TFM42	
2515	012072	012746	000001		MOV	#TFM42,-(SP)
2516	012076	010600			MOV	#1,-(SP)
2517	012100	104417			MOV	SP,R0
2518	012102	062706	000004		TRAP	C\$PNTF
2519	012106				ADD	#4,SP
2520	012106	012746	006071	PRINTB	#STM	
2521	012112	012746	000001		MOV	#STM,-(SP)
2522	012116	010600			MOV	#1,-(SP)
2523	012120	104414			MOV	SP,R0
					TRAP	C\$PNTB

2524	012122	062706	000004		ADD	#4,SP
2525	012126			ENDMSG		
2526	012126			L10026:		
2527	012126	104423		TRAP	CSMSG	
2528						
2529	012130			BGNMSG	ERR43	
2530	012130			ERR43::		
2531	012130			PRINTF	#TFM43,R5,R4	
2532	012130	010446		MOV	R4,-(SP)	
2533	012132	010546		MOV	R5,-(SP)	
2534	012134	012746	005035	MOV	#TFM43,-(SP)	
2535	012140	012746	000003	MOV	#3,-(SP)	
2536	012144	010600		MOV	SP,R0	
2537	012146	104417		TRAP	CSPNTF	
2538	012150	062706	000010	ADD	#10,SP	
2539	012154			PRINTB	#STM	
2540	012154	012746	006071	MOV	#STM,-(SP)	
2541	012160	012746	000001	MOV	#1,-(SP)	
2542	012164	010600		MOV	SP,R0	
2543	012166	104414		TRAP	CSPNTB	
2544	012170	062706	000004	ADD	#4,SP	
2545	012174			ENDMSG		
2546	012174			L10027:		
2547	012174	104423		TRAP	CSMSG	
2548	012176			BGNMSG	ERR44	
2549	012176			ERR44::		
2550	012176			PRINTF	#TFM44,#TMMC,R4	
2551	012176	010446		MOV	R4,-(SP)	
2552	012200	012746	005301	MOV	#TMMC,-(SP)	
2553	012204	012746	005120	MOV	#TFM44,-(SP)	
2554	012210	012746	000003	MOV	#3,-(SP)	
2555	012214	010600		MOV	SP,R0	
2556	012216	104417		TRAP	CSPNTF	
2557	012220	062706	000010	ADD	#10,SP	
2558	012224			PRINTB	#STM	
2559	012224	012746	006071	MOV	#STM,-(SP)	
2560	012230	012746	000001	MOV	#1,-(SP)	
2561	012234	010600		MOV	SP,R0	
2562	012236	104414		TRAP	CSPNTB	
2563	012240	062706	000004	ADD	#4,SP	
2564	012244			ENDMSG		
2565	012244			L10030:		
2566	012244	104423		TRAP	CSMSG	
2567	012246			BGNMSG	ERR45	
2568	012246			ERR45::		
2569	012246			PRINTF	#TFM45,#TMMC,R4	
2570	012246	010446		MOV	R4,-(SP)	
2571	012250	012746	005301	MOV	#TMMC,-(SP)	
2572	012254	012746	005157	MOV	#TFM45,-(SP)	
2573	012260	012746	000003	MOV	#3,-(SP)	
2574	012264	010600		MOV	SP,R0	
2575	012266	104417		TRAP	CSPNTF	
2576	012270	062706	000010	ADD	#10,SP	
2577	012274			PRINTB	#STM	
2578	012274	012746	006071	MOV	#STM,-(SP)	
2579	012300	012746	000001	MOV	#1,-(SP)	

2580	012304	010600		MOV	SP,R0
2581	012306	104414		TRAP	CSPNTB
2582	012310	062706	000004	ADD	#4,SP
2583	012314			ENDMSG	
2584	012314			L10031:	
2585	012314	104423		TRAP	CMSG
2586	012316			BGNMSG	ERR46
2587	012316			ERR46::	
2588	012316			PRINTF	#TFM46,\$GDDAT,R4
2589	012316	010446		MOV	R4,-(SP)
2590	012320	013746	002414	MOV	\$GDDAT,-(SP)
2591	012324	012746	005215	MOV	#TFM46,-(SP)
2592	012330	012746	000003	MOV	#3,-(SP)
2593	012334	010600		MOV	SP,R0
2594	012336	104417		TRAP	CSPNTF
2595	012340	062706	000010	ADD	#10,SP
2596	012344			PRINTB	#STM
2597	012344	012746	006071	MOV	#STM,-(SP)
2598	012350	012746	000001	MOV	#1,-(SP)
2599	012354	010600		MOV	SP,R0
2600	012356	104414		TRAP	CSPNTB
2601	012360	062706	000004	ADD	#4,SP
2602	012364			ENDMSG	
2603	012364			L10032:	
2604	012364	104423		TRAP	CMSG
2605					

CZKMAO KMC11-B STATIC PART2
CZKMA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 L⁵ PAGE 63
REPORT CODING SECTION

SEQ 0063

2606
2607
2608

.SBTTL REPORT CODING SECTION

CZKMAO KMC11-B STATIC PART2
CZKMA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 ^{M 5} PAGE 64
REPORT CODING SECTION

SEQ 0064

2609

;++

2610
2611
2612
2613
2614 012366
2615 012366
2616
2617
2618 012366
2619 012366 000167
2620 012370 000000
2621

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

LSRPT:: BGNRPT

EXIT RPT
.WORD JSJMP
.WORD L10033-2-

2622				
2623	012372			ENDRPT
2624	012372		L10033:	
2625	012372	104425		TRAP CSRPT
2626				
2627	012374			BGNPROT
2628	012374		L\$PROT::	
2629	012374	177777		-1
2630	012376	177777		-1
2631	012400	177777		-1
2632	012402			ENDPROT
2633				
2634	012402			BGNAUTO
2635	012402		L\$AUTO::	
2636	012402			ENDAUTO
2637	012402		L10035:	
2638	012402	104461		TRAP C\$AUTO

CZKMAO KMC11-B STATIC PART2
CZKMA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 C 6
REPORT CODING SECTION PAGE 67

SEQ 0067

2639
2640

CZKMAO KMC11-B STATIC PART2
CZKMA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 ^{D 6} PAGE 68
INITIALIZE SECTION

SEQ 0068

2641

.SBTTL INITIALIZE SECTION

2642
2643
2644
2645
2646
2647
2648 012404
2649 012404
2650
2651
2652 012404 012705 002670
2653
2654 012410 010637 002306
2655 012414 005737 002424
2656 012420 001011
2657 012422 013737 000004 002426
2658 012430 013737 000006 002430
2659 012436 012737 000001 002424
2660 012444 013737 002426 000004
2661 012452 013737 002430 000006
2662
2663
2664 012460
2665 012460 012700 000040
2666 012464 104447
2667 012466
2668 012466 103414
2669
2670 012470
2671 012470 012700 000035
2672 012474 104447
2673 012476
2674 012476 103410

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

BGNINIT
L\$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
MOV #1,FTIME
1\$: MOV SAVE4,@#4
MOV SAVE6,@#6
:SEE IF PROGRAM JUST STARTED, BR IF YES
READEF #EF.START
MOV #EF.START,R0
TRAP CSREFG
BCOMPLETE NEWST
BCS NEWST
:SEE IF THIS IS A NEW PASS, BR IF YES
READEF #EF.NEW
MOV #EF.NEW,R0
TRAP CSREFG
BCOMPLETE NEWST
BCS NEWST

2675	012500				READF	#EF.CONTINUE
2676	012500	012700	000036		MOV	#EF.CONTINUE,R0
2677	012504	104447			TRAP	CSREFG
2678	012506				BCOMPLETE	ENDIT
2679	012506	103576			BCS	ENDIT
2680						
2681					:SEE IF PROGRAM JUST RESTARTED, BR IF NOT	
2682	012510				READF	#EF.RESTART
2683	012510	012700	000037		MOV	#EF.RESTART,R0
2684	012514	104447			TRAP	CSREFG
2685	012516				BNCOMPLETE	GETPRM
2686	012516	103003			BCC	GETPRM
2687						
2688	012520				NEWST:	
2689					:RESET LOGICAL DEVICE TO -1	
2690	012520	012737	177777	002304	MOV	#-1,LOGDEV
2691						
2692					:GET UNIBUS ADRS, VECTOR, PRIORITY LEVEL, LINE UNIT,SWITCH	
2693					:PACKS, TEST CONNECTOR INFO. FOR THIS M8200,4,6,7 (CURRENT LOGICAL	
2694					:DEVICE).	
2695	012526				GETPRM:	
2696	012526	005237	002304		INC	LOGDEV
2697	012532				GPHARD	LOGDEV,R1
2698	012532	013700	002304		MOV	LOGDEV,R0
2699	012536	104442			TRAP	CSGPHRD
2700	012540	010001			MOV	R0,R1
2701	012542				BNCOMPLETE	GETPRM
2702	012542	103371			BCC	GETPRM
2703	012544	012737	000006	002356	MOV	#6,WTYPE
2704	012552	005721			TST	(R1)+
2705					:GET ADDRESS OF M8200,4,6,7	
2706	012554	011137	002456		MOV	(R1),KMCSR
2707					:GET POINTER TO M8200,4,6,7 CSR HI BYTE	
2708	012560	011137	002460		MOV	(R1),KMCSRH
2709	012564	005237	002460		INC	KMCSRH
2710					:GET POINTER TO M8200,4,6,7 CTL OUT REG	
2711	012570	011137	002462		MOV	(R1),KMCTL
2712	012574	062737	000002	002462	ADD	#2,KMCTL
2713					:GET POINTER TO M8200,4,6,7 PORT REG - SEL 4	
2714	012602	011137	002464		MOV	(R1),KMPO4
2715	012606	062737	000004	002464	ADD	#4,KMPO4
2716					:GET POINTER TO M8200,4,6,7 PORT REG - SEL 6	
2717	012614	012137	002466		MOV	(R1)+,KMPO6
2718	012620	062737	000006	002466	ADD	#6,KMPO6
2719					:GET POINTER TO RCV VECTOR	
2720	012626	011137	002446		MOV	(R1),KMRVEC
2721					:GET POINTER TO RCV PRIORITY LEVEL	
2722	012632	011137	002450		MOV	(R1),KMRLVL
2723	012636	062737	000002	002450	ADD	#2,KMRLVL
2724					:GET POINTER TO TX VECTOR	
2725	012644	011137	002452		MOV	(R1),KMTVEC
2726	012650	062737	000004	002452	ADD	#4,KMTVEC
2727					:GET POINTER TO TX PRIORITY LEVEL	
2728	012656	011137	002454		MOV	(R1),KMTLVL
2729	012662	062737	000006	002454	ADD	#6,KMTLVL
2730					:PUT VECTOR INTO STAT1	

```
2731 012670 012137 002440      MOV      (R1)+,STAT1
2732      :PUT PRIORITY INTO STAT1
2733 012674 052137 002440      BIS      (R1)+,STAT1
2734      :SEE IF NO LINE UNIT, SET BIT IF YES
2735 012700 005711      TST      (R1)
2736 012702 001004      BNE      50000$
2737 012704 052737 010000 002440  BIS      #BIT12,STAT1
2738 012712 000416      BR       4$
2739 012714      50000$:
2740      :SEE IF M8201 LINE UNIT, SET BIT IF YES
2741 012714 021127 000001      CMP      (R1),#1
2742 012720 001001      BNE      50001$
2743 012722 000412      BR       4$
2744 012724      50001$:
2745      :SEE IF M8202 LINE UNIT, SET BIT IF YES
2746 012724 021127 000002      CMP      (R1),#2
2747 012730 001004      BNE      50002$
2748 012732 052737 020000 002440  BIS      #BIT13,STAT1
2749 012740 000403      BR       4$
2750 012742      50002$:
2751      :SET BIT FOR M8203 LINE UNIT
2752 012742 052737 100000 002440  BIS      #BIT15,STAT1
2753 012750      4$:
2754      :SET BIT IN STAT1 FOR TEST CONNECTOR
2755 012750 056137 000010 002440  BIS      10(R1),STAT1
2756 012756 005037 002432      CLR      SKIP33
2757 012762 005761 000012      TST      12(R1)
2758 012766 001403      BEQ      45$
2759 012770 012737 177777 002432  MOV      #-1,SKIP33
2760 012776 062701 000002      45$: ADD      #2,R1
2761      :SET SWITCH PACK #1 IN STAT2 LOW BYTE
2762 013002 012137 002442      MOV      (R1)+,STAT2
2763      :SET SWITCH PACK #2 IN STAT2 HIGH BYTE
2764 013006 111137 002443      MOV      (R1),STAT2+1
2765
2766      :INCREMENT LOGICAL UNIT (DEVICE) NUMBER
2767      :
2768 013012 000240      INC      LOGDEV
2769 013014 000240      NOP
2770      NOP
2771 013016 012737 002000 002400  MOV      #2000,MEMSZ
2772 013024 005037 002374      CLR      TYPE
2773 013030 123727 002356 000000  CMPB     WTYPE,#0
2774 013036 001422      BEQ      ENDIT
2775 013040 123727 002356 000004  CMPB     WTYPE,#4      :KMC?
2776 013046 001004      BNE      5$
2777 013050 012737 000001 002374  MOV      #1,TYPE
2778 013056 000412      BR       ENDIT
2779 013060 012737 007777 002400  5$: MOV      #7777,MEMSZ
2780 013066 123727 002356 000006  CMPB     WTYPE,#6
2781 013074 001003      BNE      ENDIT
2782 013076 012737 000001 002374  MOV      #1,TYPE
2783 013104      ENDIT:
2784 013104      ENDINIT
2785 013104      L10036:
2786 013104 104411      TRAP     CSINIT
```

CZKMAO KMC11-B STATIC PART2
CZKMA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 H 6
INITIALIZE SECTION PAGE 72

SEQ 0072

2787
2788
2789
2790
2791
2792
2793

.EVEN

2794
2795
2796
2797
2798
2799
2800
2801 013106
2802 013106
2803
2804
2805 013106
2806 013106
2807 013106 104412
2808
2809
2810
2811
2812

.SBTTL CLEANUP CODING SECTION

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

BGNCLN
L\$CLEAN::

ENDCLN
L10037: TRAP C\$CLEAN

2813
2814
2815
2816
2817
2818
2819
2820 013110
2821 013110
2822
2823 013110
2824 013110 104433
2825 013112
2826 013112
2827 013112 104453
2828
2829
2830
2831
2832

```
.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  
L10040:  
          TRAP   CSDU
```

2833
2834
2835
2836
2837
2838
2839
2840
2841 013114
2842 013114
2843 013114
2844 013114
2845 013114 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
L10041: TRAP CSAU


```
2846 .SBTTL HARDWARE TESTS
2847
2848
2849
2850 ;START OF CODE BLOCK WHICH IS USED AS DATA
2851 013116 ROMMAP:
2852
2853 013116 BADHEAD
2854 :***** TEST 1 *****
2855 :*VERIFY THAT REFERENCING UNIBUS DEVICE REGISTERS
2856 :*DOES NOT CAUSE A TIME OUT TRAP
2857 013116 BADHEAD
2858 :***** TEST 1 *****
2859
2860 013116 BGNTST
2861 013116 T1::
2862 013116 013701 002456 MOV KMCSR,R1 ;R1 CONTAINS BASE M8200,4,6,7 ADDRESS
2863 013122 012705 000004 MOV #4,R5 ;4 REGISTERS TO BE TESTED
2864 013126 012737 013164 000004 MOV #2$,4 ;SET OUT TIMEOUT TRAP
2865 013134 012737 000240 000006 MOV #240,6 ;LEVEL 7
2866 013142 005711 1$: TST (R1) ;REFERENCE DEVICE REGISTERS
2867 013144 000240 NOP
2868 013146 ESCAPE TST
2869 013146 104410 TRAP C$ESCAPE
2870 013150 000054 .WORD L10042-.
2871 013152 062701 000002 ADD #2,R1 ;NEXT REGISTER
2872 013156 005305 DEC R5 ;DEC REGISTER COUNT
2873 013160 001370 BNE 1$ ;BR IF NOT LAST REGISTER
2874 013162 000410 BR 3$
2875
2876 013164 062706 000004 2$: ADD #4,SP
2877 013170 ERROR 36 ;TIME OUT ERROR
2878 013174 104455 TRAP C$ERDF
2879 013176 000044 .WORD 36
2880 013200 005425 .WORD EMO
2881 013202 011706 .WORD ERR36
2882
2883 013204 013737 002426 000004 3$: MOV SAVE4,4
2884 013212 013737 002430 000006 MOV SAVE6,6
2885 013220 ESCAPE TST
2886 013220 104410 TRAP C$ESCAPE
2887 013222 000002 .WORD L10042-.
2888
2889 013224 ENDTST
2890 013224 L10042:
2891 013224 104401 TRAP C$ETST
2892 .EVEN
```

```
2893 013226          BADHEAD
2894                :***** TEST 2 *****
2895                :*TEST OF BR RIGHT SHIFT
2896                :*VERIFY THAT A DEST OF BR RSH (011) OF A MICRO-INSTRUCTION
2897                :*SHIFTS THE RESULTING BR DATA RIGHT ONCE.
2898 013226          BADHEAD
2899                :***** TEST 2 *****
2900
2901 013226          BGNTST
2902 013226          T2::
2903
2904 013226          MSTCLR          :R1 CONTAINS BASE M8200,4,6,7 ADDRESS
2905 013232 013701 002456  MOV      KMCSR,R1          :MASTER CLEAR M8200,4,6,7
2906 013236 005011      CLR      (R1)           :R1 = M8200,4,6,7 BASE ADDRESS
2907 013240 012705 052525  MOV      #52525,R5        :CLEAR SELO
2908 013244 010561 000004  MOV      R5,4(R1)       :START WITH 125
2909 013250          ROMCLK          :PORT4 125
2910 013254 120500      120500      :NEXT WORD IS INSTRUCTION
2911 013256          ROMCLK          :PORT4 TO BR-REG
2912 013262 061620      061620      :NEXT WORD IS INSTRUCTION
2913 013264          ROMCLK          :BR RSH BR, SHIFT BR RIGHT
2914 013270 061225      061225      :NEXT WORD IS INSTRUCTION
2915 013272 006005          ROR      R5           :PORT5 BR
2916 013274 116104 000005  MOVB    5(R1),R4        :R5 = "EXPECTED"
2917 013300 120504          CMPB    R5,R4          :R4 = "FOUND"
2918 013302 001410      BEQ     1$            :DID BR SHIFT RIGHT ONCE?
2919 013304          ERROR    12          :BR IF YES
2920 013310 104455      TRAP    C$ERDF        :BR RIGHT SHIFT ERROR
2921 013312 000014      .WORD  12
2922 013314 005425      .WORD  EMO
2923 013316 010212      .WORD  ERR12
2924
2925 013320          ESCAPE TST          :SHOULD BE 52
2926 013320 104410      TRAP    C$ESCAPE
2927 013322 000074      .WORD  L10043-
2928 013324          1$:
2929 013324          ROMCLK          :NEXT WORD IS INSTRUCTION
2930 013330 061620      061620      :BR RSH BR, SHFT BR RIGHT AGAIN
2931 013332          ROMCLK          :NEXT WORD IS INSTRUCTION
2932 013336 061225      061225      :PORT5 BR
2933 013340 006005          ROR      R5           :R5 = "EXPECTED"
2934 013342 116104 000005  MOVB    5(R1),R4        :R4 = "FOUND"
2935 013346 120504          CMPB    R5,R4          :DID BR SHIFT RIGHT?
2936 013350 001406      BEQ     2$            :BR IF YES
2937 013352          ERROR    12          :BR RIGHT SHIFT ERROR
2938 013356 104455      TRAP    C$ERDF
2939 013360 000014      .WORD  12
2940 013362 005425      .WORD  EMO
2941 013364 010212      .WORD  ERR12
2942
2943 013366          2$:
2944 013366          CHKREG
2945          .LIST    ME
2946 013366 004737 003014  JSR     PC,RDKM        : READ KMC11 REGISTERS
2947 013372 012702 013404  MOV     #64$,R2        : LOAD POINTER
2948 013376 004737 003120  JSR     PC,RCHECK     : CHECK FOR NO CHANGE
```

2949	013402	000405				BR	65\$: SKIP LIST
2950	013404	000	000	000	64\$:	.BYTE20		
2951	013407	000	000	000					
2952	013412	000	000	000					
2953	013415	020							
2954	013416				65\$:				
2955	013416				ENDTST				
2956	013416				L10043:				
2957	013416	104401			TRAP	CSETST			


```

2958 013420          BADHEAD
2959                ;***** TEST 3 *****
2960                ;*IOP CRAM WRITE/READ TEST
2961                ;*FLOAT A 1 THROUGH EACH CRAM LOCATION
2962 013420          BADHEAD
2963                ;***** TEST 3 *****
2964
2965 013420          BGNTST
2966 013420          T3::
2967 013420
2968 013424          MYINT
2969                MSTCLR
2970 013430 005037 002376          CLR      MRO          ;R1 CONTAINS BASE M8200,4,6,7 ADDRESS
2971 013434 012702 000001          ADR4:  MOV      #1,R2      ;MRO = CRAM ADDRESS
2972 013440          ADR5:
2973 013440          BGNSEG
2974 013440 104404          TRAP     CSBSEG
2975 013442 012711 002000          3$:  MOV      #BIT10,(R1)    ;SET ROMO
2976 013446 013761 002376 000004  MOV      MRO,4(R1)      ;WRITE ADDRESS TO SEL4
2977 013454 010261 000006          MOV      R2,6(R1)      ;LOAD SEL6 WITH WRITE DATA
2978 013460 052711 020000          BIS      #BIT13,(R1)   ;WRITE SEL6 INTO CRAM
2979 013464 016104 000006          MOV      6(R1),R4      ;READ CRAM INTO 'FOUND'
2980 013470 020204          CMP      R2,R4         ;IS DATA CORRECT?
2981 013472 001410          BEQ     4$            ;BR IF OK
2982 013474          ERROR 1          ;ERROR
2983 013500 104455          TRAP     C$ERDF
2984 013502 000001          .WORD   1
2985 013504 005425          .WORD   EMO
2986 013506 006274          .WORD   ERR1
2987 013510          ESCAPE SEG
2988 013510 104410          TRAP     C$ESCAPE
2989 013512 000002          .WORD   10000$-.
2990 013514          4$:
2991 013514 10000$:          ENDSEG
2992 013514 104405          TRAP     C$ESEG
2993 013516 000241          CLC
2994 013520 006102          ROL     R2            ;CLEAR CARRY
2995 013522 001346          BNE     ADR5          ;SHIFT WRITE DATA
2996 013524 005237 002376          INC     MRO           ;BSR IF NOT DONE THIS ADDRESS
2997 013530 023737 002400 002376          CMP     MEMSZ,MRO     ;BUMP TO NEXT CRAM ADDRESS
2998 013536 001336          BNE     ADR4          ;DONE YET?
2999 013540          5$:
3000 013540          CHKREG
3001          .LIST  ME
3002 013540 004737 003014          JSR     PC,RDKM       ; READ KMC11 REGISTERS
3003 013544 012702 013556          MOV     #64$,R2      ; LOAD POINTER
3004 013550 004737 003120          JSR     PC,RCHECK     ; CHECK FOR NO CHANGE
3005 013554 000405          BR      65$          ; SKIP LIST
3006 013556          000          000          000 64$:  .BYTE  .....20
3007 013561          000          000          000
3008 013564          000          000          000
3009 013567          020
3010 013570          65$:
3011 013570          ENDTST
3012 013570          L10044:
3013 013570 104401          TRAP     C$SETST

```

```
3014 013572          BADHEAD
3015                :***** TEST 4 *****
3016                :*IOP CRAM WRITE/READ TEST
3017                :*FLOAT A 0 THROUGH EACH CRAM LOCATION
3018 013572          BADHEAD
3019                :***** TEST 4 *****
3020
3021 013572          BGNTST
3022 013572          T4::
3023 013572
3024 013576          MYINT
3025 013602 005037 002376          MSTCLR          :MASTER CLEAR M8200,4,6,7
3026 013606 012702 000001          CLR MRO          :MRO = CRAM ADDRESS
3027 013612          ADR1: MOV #1,R2          :R2 = WRITE DATA
3028 013612          ADR2:
3029 013612 104404          BGNSEG
3030 013614 005102          TRAP CSBSEG
3031 013616 012711 002000          COM R2          :MAKE IT A FLOATING ZERO
3032 013622 013761 002376 000004 3$: MOV #BIT10,(R1) :SET ROMO
3033 013630 010261 000006          MOV MRO,4(R1)   :WRITE ADDRESS TO SEL4
3034 013634 052711 020000          MOV R2,6(R1)   :LOAD SEL6 WITH WRITE DATA
3035 013640 016104 000006          BIS #BIT13,(R1) :WRITE SEL6 INTO CRAM
3036 013644 020204          MOV 6(R1),R4   :READ CRAM INTO 'FOUND'
3037 013646 001410          CMP R2,R4      :IS DATA CORRECT?
3038 013650          BEQ 4$
3039 013654 104455          ERROR 1      :BR IF OK
3040 013656 000001          TRAP CSERDF   :ERROR
3041 013660 005425          .WORD 1
3042 013662 006274          .WORD EMO
3043 013664          .WORD ERR1
3044 013664 104410          ESCAPE SEG
3045 013666 000002          TRAP C$ESCAPE
3046 013670          .WORD 10000$-.
3047 013670          4$: ENDSEG
3048 013670 104405          10000$: TRAP C$ESEG
3049 013672 005102          COM R2
3050 013674 000241          CLC          :BACK TO FLOATING ONE
3051 013676 006102          ROL R2      :CLEAR CARRY
3052 013700 001344          BNE ADR2     :SHIFT WRITE DATA
3053 013702 005237 002376          INC MRO      :BR IF NOT DONE THIS ADDRESS
3054 013706 023737 002400 002376          CMP MEMSZ,MRO :BUMP TO NEXT CRAM ADDRESS
3055 013714 001334          BNE ADR1     :DONE YET?
3056 013716          5$:
3057 013716
3058
3059 013716 004737 003014          CHKREG
3060 013722 012702 013734          .LIST ME
3061 013726 004737 003120          JSR PC,RDKM   : READ KMC11 REGISTERS
3062 013732 000405          MOV #64$,R2  : LOAD POINTER
3063 013734 000 000 000 64$: JSR PC,RCHECK : CHECK FOR NO CHANGE
3064 013737 000 000 000          BR 65$       : SKIP LIST
3065 013742 000 000 000          .BYTE .....20
3066 013745 020
3067 013746          65$:
3068 013746          ENDTST
3069 013746          L10045:
```

CZKMAO KMC11-B STATIC PART2
CZKMA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50^{D 7} PAGE 81
HARDWARE TESTS

SEQ 0081

3070 013746 104401

TRAP CSETST


```
3071 013750          BADHEAD
3072                :***** TEST 5 *****
3073                :*IOP CRAM DUAL ADDRESSING TEST
3074                :*WRITE EACH ADDRESS INTO ITSELF, READ EACH
3075                :*ADDRESS TO VERIFY CORRECT ADDRESSING
3076 013750          BADHEAD
3077                :***** TEST 5 *****
3078
3079 013750          BGNTST
3080 013750          T5::
3081 013750
3082
3083 013754          MYINT
3084 013760 005037 002376  MSTCLR                ;R1 CONTAINS BASE M8200,4,6,7 ADDRESS
3085 013764          CLR MRO                ;MASTER CLEAR M8200,4,6,7
3086 013764 104404  TRAP C$BSEG            ;MRO =CRAM ADDRESS
3087 013766 013702 002376 1$:  MOV MRO,R2                ;SAVE R2 FOR TYPEOUT
3088 013772 012711 002000  MOV #BIT10,(R1)         ;SET ROMO
3089 013776 013761 002376 000004  MOV MRO,4(R1)           ;WRITE ADDRESS TO SEL4
3090 014004 013761 002376 000006  MOV MRO,6(R1)           ;LOAD SEL6 WITH WRITE DATA
3091 014012 052711 020000  BIS #BIT13,(R1)        ;WRITE CRAM
3092 014016          SKIP06 15$             ;IF M8206,SKIP NEXT INSTR.
3093 014026 005061 000006  CLR 6(R1)                ;CLEAR SEL 6
3094 014032
3095 014032 016104 000006 15$:  MOV 6(R1),R4            ;SHOULD READ BACK OWN ADDRESS
3096 014036 023704 002376  CMP MRO,R4              ;IS DATA CORRECT?
3097 014042 001410  BEQ 2$                  ;BR IF YES
3098 014044          ERROR 1                ;DATA ERROR
3099 014050 104455  TRAP C$ERDF
3100 014052 000001  .WORD 1
3101 014054 005425  .WORD EMO
3102 014056 006274  .WORD ERR1
3103 014060          ESCAPE SEG
3104 014060 104410  TRAP C$ESCAPE
3105 014062 000002  .WORD 10000$-
3106 014064          ENDSEG
3107 014064          2$:
10000$:
3108 014064 104405  TRAP C$ESEG
3109 014066          BGNSEG
3110 014066 104404  TRAP C$BSEG
3111 014070 005237 002376  INC MRO                ;BUMP TO NEXT ADDRESS
3112 014074 023737 002400 002376  CMP MEMSZ,MRO          ;DONE WRITING YET?
3113 014102 001331  BNE 1$                  ;BR IF NO
3114 014104 005037 002376  CLR MRO                ;RESTART AT ADDRESS 0
3115 014110 013702 002376 3$:  MOV MRO,R2                ;SAVE R2 FOR TYPEOUT
3116 014114 012711 002000  MOV #BIT10,(R1)         ;SET ROMO
3117 014120 013761 002376 000004  MOV MRO,4(R1)           ;SEL4 = CRAM ADDRESS
3118 014126 016104 000006  MOV 6(R1),R4            ;READ CRAM INTO 'FOUND'
3119 014132 023704 002376  CMP MRO,R4              ;IS DATA CORRECT?
3120 014136 001411  BEQ 4$                  ;BR IF YES
3121 014140          ERROR 2                ;DUAL ADDRESSING ERROR
3122 014144 104455  TRAP C$ERDF
3123 014146 000002  .WORD 2
3124 014150 005425  .WORD EMO
3125 014152 006452  .WORD ERR2
3126 014154          ESCAPE SEG
```

3127	014154	104410					TRAP	C\$ESCAPE	
3128	014156	000002					.WORD	10001\$-	
3129	014160						ENDSEG		
3130	014160					10001\$:			
3131	014160	104405					TRAP	C\$ESEG	
3132	014162					4\$:			:LOOP TO 3\$ IF SW09=1
3133	014162	005237	002376				INC	MRO	:BUMP TO NEXT ADDRESS
3134	014166	023737	002400	002376			CMP	MEMSZ,MRO	:DONE WRITING YET?
3135	014174	001345					BNE	3\$:BR IF NO
3136	014176					5\$:			
3137	014176						CHKREG		
3138							.LIST	ME	
3139	014176	004737	003014				JSR	PC,RDKM	: READ KMC11 REGISTERS
3140	014202	012702	014214				MOV	#64\$,R2	: LOAD POINTER
3141	014206	004737	003120				JSR	PC,RCHECK	: CHECK FOR NO CHANGE
3142	014212	000405					BR	65\$: SKIP LIST
3143	014214	000	000	000	64\$:		.BYTE20	
3144	014217	000	000	000					
3145	014222	000	000	000					
3146	014225	020							
3147	014226					65\$:			
3148	014226					ENDTST			
3149	014226					L10046:			
3150	014226	104401					TRAP	C\$ETST	

```
3151 014230          BADHEAD
3152                :***** TEST 6 *****
3153                :*IOP MAIN MEMORY TEST
3154                :*FLOAT A 1 THROUGH ALL MAIN MEMORY LOCATIONS
3155 014230          BADHEAD
3156                :***** TEST 6 *****
3157
3158 014230          BGNTST
3159 014230          T6::
3160 014230          MYINT
3161
3162 014234          MSTCLR          :R1 CONTAINS BASE M8200,4,6,7 ADDRESS
3163 014240 005037 002350          CLR          :MASTER CLEAR M8200,4,6,7
3164 014244 012737 000001 002376 1$: MOV          :START WITH ADDRESS 0
3165 014252 042737 003777 014306 33$: BIC          :START WITH BIT 0
3166 014260 042737 000037 014314          BIC          :CLEAR ADDRESS FIELD OF INSTRUCTION
3167 014266 153737 002350 014306          BISB         :CLEAR ADDRESS FIELD OF INSTRUCTION
3168 014274 153737 002351 014314          BISB         :ADD ADDRESS TO INSTRUCTION7
3169 014302          ROMCLK          :ADD ADDRESS TO INSTRUCTION
3170 014306 010000          66$: 010000          :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3171 014310          ROMCLK
3172 014314 004000          68$: 004000          :LOAD MAR HI
3173 014316 013761 002376 000004          MOV          MRO,4(R1) :WRITE PATTERN IN PORT4
3174 014324          ROMCLK          :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3175 014330 122500          122500          :MOVE PORT4 TO MEMORY
3176 014332          ROMCLK          :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3177 014336 040620          040620          :MOVE MEMORY TO BR
3178 014340          ROMCLK          :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3179 014344 061225          61225          :MOVE BR TO PORT5
3180 014346 013705 002376          MOV          MRO,R5 :PUT 'EXPECTED' IN R5
3181 014352 116104 000005          MOVB         5(R1),R4 :PUT 'FOUND' IN R4
3182 014356 120504          CMPB         R5,R4 :DATA CORRECT?
3183 014360 001410          BEQ          67$ :BR IF YES
3184 014362          ERROR          6 :DATA ERROR
3185 014366 104455          TRAP         C$ERDF
3186 014370 000006          .WORD        6
3187 014372 005425          .WORD        EMO
3188 014374 007332          .WORD        ERR6
3189 014376          ESCAPE TST
3190 014376 104410          TRAP         C$ESCAPE
3191 014400 000056          .WORD        L10047-.
3192 014402          67$:
3193 014402 000241          CLC
3194 014404 106137 002376          ROLB         MRO
3195 014410 001320          BNE          33$ :CLEAR CARRY
3196 014412 005237 002350          INC          :SHIFT BIT IN MRO
3197 014416 023737 002400 002350          CMP          :DONE IF MRO=0
3198 014424 001307          BNE          :NEXT ADDRESS
3199 014426          2$:
3200 014426          :LAST ADDRESS?
3201          CHKREG
3202 014426 004737 003014          .LIST        ME
3203 014432 012702 014444          JSR          PC,RDKM : READ KMC11 REGISTERS
3204 014436 004737 003120          MOV          #64$,R2 : LOAD POINTER
3205 014442 000405          JSR          PC,R$CHECK : CHECK FOR NO CHANGE
3206 014444 000 000 000 64$: .BYTE        65$ : SKIP LIST
                        .....20
```


3207	014447	000	000	000
3208	014452	000	000	000
3209	014455	020		
3210	014456			
3211	014456			
3212	014456			
3213	014456	104401		

65\$:
ENDTST
L10047:
TRAP CSETST

```
3214 014460 BADHEAD
3215 :***** TEST 7 *****
3216 :*IOP MAIN MEMORY TEST
3217 :*FLOAT A 0 THROUGH ALL MAIN MEMORY LOCATIONS
3218 014460 BADHEAD
3219 :***** TEST 7 *****
3220
3221 014460 BGNTST
3222 014460 T7::
3223 014460 MYINT
3224 :R1 CONTAINS BASE M8200,4,6,7 ADDRESS
3225 014464 MSTCLR :MASTER CLEAR M8200,4,6,7
3226 014470 005037 002350 CLR FLAG :START WITH ADDRESS 0
3227 014474 012737 000001 002376 1$: MOV #1,MRO :START WITH BIT 0
3228 014502 005137 002376 34$: COM MRO :CHANGE TO FLOATING 0
3229 014506 042737 003777 014542 35$: BIC #3777,66$ :CLEAR ADDRESS FIELD OF INSTRUCTION
3230 014514 042737 000037 014550 BIC #37,68$ :CLEAR ADDRESS FIELD OF INSTRUCTION
3231 014522 153737 002350 014542 BISB FLAG,66$ :ADD ADDRESS TO INSTRUCTION
3232 014530 153737 002351 014550 BISB FLAG+1,68$ :ADD ADDRESS TO INSTRUCTION
3233 014536 ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3234 014542 010000 66$: 010000 :LOAD MAR LO WITH ADDRESS IN FLAG
3235 014544 ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3236 014550 004000 68$: 004000 :LOAD MAR HI
3237 014552 013761 002376 000004 MOV MRO,4(R1) :WRITE PATTERN IN PORT4
3238 014560 ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3239 014564 122500 122500 :MOVE PORT4 TO MEMORY
3240 014566 ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3241 014572 040620 040620 :MOVE MEMORY TO BR
3242 014574 ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3243 014600 061225 61225 :MOVE BR TO PORT5
3244 014602 013705 002376 MOV MRO,R5 :PUT 'EXPECTED' IN R5
3245 014606 116104 000005 MOVB 5(R1),R4 :PUT 'FOUND' IN R4
3246 014612 120504 CMPB R5,R4 :DATA CORRECT?
3247 014614 001406 BEQ 67$ :BR IF YES
3248 014616 ERROR 6 :DATA ERROR
3249 014622 104455 TRAP C$ERDF
3250 014624 000006 .WORD 6
3251 014626 005425 .WORD EMO
3252 014630 007332 .WORD ERR6
3253 014632 67$: ESCAPE TST
3254 014632 104410 TRAP C$ESCAPE
3255 014634 000062 .WORD L10050-.
3256 014636 005137 002376 COM MRO :CHANGE TO FLOATING 1
3257 014642 000241 CLC :CLEAR CARRY
3258 014644 106137 002376 ROLB MRO :SHIFT BIT IN MRO
3259 014650 001314 BNE 34$ :DONE IF MRO=0
3260 014652 005237 002350 INC FLAG :NEXT ADDRESS
3261 014656 023737 002400 002350 CMP MEMSZ,FLAG :LAST ADDRESS?
3262 014664 001303 BNE 1$ :BR IF NO
3263 014666 2$:
3264 014666
3265 CHKREG
3266 014666 004737 003014 .LIST ME : READ KMC11 REGISTERS
3267 014672 012702 014704 JSR PC,RDKM : LOAD POINTER
3268 014676 004737 003120 MOV #64$,R2 : CHECK FOR NO CHANGE
3269 014702 000405 JSR PC,RCHECK : SKIP LIST
BR 65$
```

3270	014704	000	000	000	64\$:	.BYTE,20
3271	014707	000	000	000			
3272	014712	000	000	000			
3273	014715	020					
3274	014716				65\$:		
3275	014716				ENDTST		
3276	014716				L10050:		
3277	014716	104401			TRAP	CSETST	


```
3278 014720 BADHEAD
3279 :***** TEST 8 *****
3280 :*IOP MAIN MEMORY DUAL ADDRESSING TEST
3281 :*LOAD EACH MEMORY LOCATION WITH ITS OWN ADDRESS
3282 :*READ BACK EACH LOCATION TO VERIFY CORRECT ADDRESSING
3283 014720 BADHEAD
3284 :***** TEST 8 *****
3285
3286 014720 BGNTST
3287 014720 T8::
3288 014720
3289 MYINT
3290 014724 MSTCLR ;R1 CONTAINS BASE M8200,4,6,7 ADDRESS
3291 014730 005037 002350 CLR ;MASTER CLEAR M8200,4,6,7
3292 014734 013702 002350 1$: MOV FLAG,R2 ;START WITH ADDRESS 0
3293 014740 042737 003777 014774 BIC #3777,2$ ;PUT DATA IN R2
3294 014746 042737 000037 015002 BIC #37,7$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
3295 014754 153737 002350 014774 BISB FLAG,2$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
3296 014762 153737 002351 015002 BISB FLAG+1,7$ ;ADD ADDRESS TO INSTRUCTION
3297 014770 ROMCLK ;ADD ADDRESS TO INSTRUCTION
3298 014774 010000 2$: 010000 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3299 014776 ROMCLK ;LOAD MAR LO
3300 015002 004000 7$: 004000 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3301 015004 010261 000004 MOV R2,4(R1) ;LOAD MAR HI
3302 015010 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3303 015014 122500 ROMCLK 122500 ;MOVE PORT4 TO MEMORY
3304 015016 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3305 015022 040620 ROMCLK 040620 ;MOVE MEMORY TO THE BR
3306 015024 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3307 015030 061225 ROMCLK 61225 ;MOVE BR TO PORT5
3308 015032 010205 MOV R2,R5 ;PUT 'EXPECTED' IN R5
3309 015034 116104 000005 MOVB 5(R1),R4 ;PUT 'FOUND' IN R4
3310 015040 120504 CMPB R5,R4 ;DATA CORRECT?
3311 015042 001406 BEQ 3$ ;BR IF YES
3312 015044 ERROR 6 ;DATA ERROR
3313 015050 104455 TRAP C$ERDF
3314 015052 000006 .WORD 6
3315 015054 005425 .WORD EMO
3316 015056 007332 .WORD ERR6
3317 015060 3$: ESCAPE TST
3318 015060 104410 TRAP C$ESCAPE
3319 015062 000204 .WORD L10051-
3320 015064 005237 002350 INC FLAG ;NEXT ADDRESS
3321 015070 023737 002400 002350 CMP MEMSZ,FLAG ;LAST ADDRESS?
3322 015076 001316 BNE 1$ ;BR IF NO
3323 015100 012737 015112 002302 MOV #4$,LOCK ;NEW SCOPE 1
3324 015106 005037 002350 CLR FLAG ;RESTART AT ADDRESS 0
3325 015112 013702 002350 4$: MOV FLAG,R2 ;PUT DATA IN R2
3326 015116 042737 003777 015144 BIC #3777,5$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
3327 015124 042737 000037 015152 BIC #37,8$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
3328 015132 153737 002350 015144 BISB FLAG,5$ ;ADD ADDRESS TO INSTRUCTION
3329 015140 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC+5304
3330 015144 010000 5$: 010000 ;LOAD THE MAR LO
3331 015146 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3332 015152 004000 8$: 004000 ;LOAD MAR HI
3333 015154 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

3334	015160	040620				040620		:MOVE MEMORY TO THE BR
3335	015162					ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3336	015166	061225				61225		:MOV BR TO PORT5
3337	015170	010205				MOV	R2,R5	:PUT 'EXPECTED' IN R5
3338	015172	116104	000005			MOVB	5(R1),R4	:PUT 'FOUND' IN R4
3339	015176	120504				CMPB	R5,R4	:DATA CORRECT?
3340	015200	001406				BEQ	6\$:BR IF YES
3341	015202					ERROR	6	:ADDRESSING ERROR
3342	015206	104455				TRAP	C\$ERDF	
3343	015210	000006				.WORD	6	
3344	015212	005425				.WORD	EMO	
3345	015214	007332				.WORD	ERR6	
3346	015216				6\$:	ESCAPE	TST	
3347	015216	104410				TRAP	C\$ESCAPE	
3348	015220	000046				.WORD	L10051-	
3349	015222	005237	002350			INC	FLAG	:NEXT ADDRESS
3350	015226	023737	002400	002350		CMP	MEMSZ,FLAG	:IS IT THE LAST
3351	015234	001326				BNE	4\$:BR IF NO
3352	015236				9\$:			
3353	015236					CHKREG		
3354						.LIST	ME	
3355	015236	004737	003014			JSR	PC,RDKM	: READ KMC11 REGISTERS
3356	015242	012702	015254			MOV	#64\$,R2	: LOAD POINTER
3357	015246	004737	003120			JSR	PC,RCHECK	: CHECK FOR NO CHANGE
3358	015252	000405				BR	65\$: SKIP LIST
3359	015254	000	000	000	64\$:	.BYTE20	
3360	015257	000	000	000				
3361	015262	000	000	000				
3362	015265	020						
3363	015266				65\$:			
3364	015266				ENDTST			
3365	015266				L10051:			
3366	015266	104401				TRAP	C\$ETST	

```

3367 015270          BADHEAD
3368                :***** TEST 9 *****
3369                :*IOP MAR TEST
3370                :*PERFORM DUAL ADDRESSING TEST
3371                :*USING MAR AUTO-INC FEATURE
3372 015270          BADHEAD
3373                :***** TEST 9 *****
3374
3375 015270          BGNTST
3376 015270          T9::
3377 015270
3378                MYINT
3379 015274          MSTCLR                :R1 CONTAINS BASE M8200,4,6,7 ADDRESS
3380 015300 005002  CLR                    :MASTER CLEAR M8200,4,6,7
3381 015302 013703 002400  MOV R2        :START WITH A ZERO
3382 015306 005203          MEMSZ,R3      :GET MEMORY SIZE
3383 015310          INC R3                :STOP ADDR=MEMSZ+1
3384 015314 010000  ROMCLK                :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3385 015316          010000              :LOAD MAR WITH A ZERO
3386 015324 010261 000004  1$: MOV R2,4(R1) :WRITE DATA TO PORT4
3387 015330          ROMCLK                :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3388 015334 136500 136500          :MEM PORT4, AUTO-INC MAR
3389 015336 005202          INC R2        :INCREMENT DATA
3390 015340 020302          CMP R3,R2   :DONE YET?
3391 015342 001370          BNE 1$       :BR IF NO
3392 015344 005002          CLR R2        :RESTART WITH A ZERO
3393 015346          ROMCLK                :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3394 015352 010000 010000          :LOAD MAR WITH A ZERO
3395 015354          CLRMAR
3396 015362          2$:
3397 015362          SRMCLK                :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3398 015366 055224 055224          :MOVE MEM TO PORT4
3399 015370 010205 010205  MOV R2,R5   :PUT 'EXPECTED' IN R5
3400 015372 116104 000004  MOVB 4(R1),R4 :PUT 'FOUND' IN R4
3401 015376 120504          CMPB R5,R4  :DATA CORRECT?
3402 015400 001406          BEQ 3$     :BR IF YES
3403 015402          ERROR 11            :MAR ERROR
3404 015406 104455          TRAP C$ERDF
3405 015410 000013          .WORD 11
3406 015412 005425          .WORD EMO
3407 015414 010040          .WORD ERR11
3408 015416          3$:
3409 015422 000000 0          SRMCLK 0          :DUMP NOP INSTR. TO CLK AUTO INC IN MAR.
3410 015424 005004          CLR R4
3411 015426          ROMCLK                :READ IBUS* <15> (MAR HIGH)
3412 015432 121325 121325          :MAR HIGH _POT 5
3413
3414 015434          ROMCLK                :READ IBUS* <14> (MAR LOW)
3415 015440 121304 121304
3416 015442 016104 000004  MOV 4(R1),R4 :ADD TO MAR HIGH.
3417 015446 042704 160000  BIC #160000,R4
3418 015452 005202          INC R2
3419 015454 020237 002400  CMP R2, MEMSZ
3420 015460 001002          BNE 35$
3421 015462 052702 010000  BIS #10000,R2 :IF AT HIGH LIMIT,ADD IN OVERFLOW BIT.
3422 015466          35$:

```



```

3423 015466 020204      CMP      R2,R4      ;ADDR. OK?
3424 015470 001406      BEQ      4$
3425 015472              ERROR     11          ;ERROR MAR ADDR. BAD IN IBUS <14>AND <15>
3426 015476 104455      TRAP     C$ERDF
3427 015500 000013      .WORD   11
3428 015502 005425      .WORD   EMO
3429 015504 010040      .WORD   ERR11
3430              ;EXPECTED (R4) IS COMBINATION OF
3431              ;IBUS* <14> AND <15>
3432 015506      4$:
3433 015506
3434 015506 104410      ESCAPE   TST
3435 015510 000136      TRAP     C$ESCAPE
3436 015512 032702 010000      .WORD   L10052-
3437 015516 001721      BIT      #10000,R2    ;DONE YET?
3438              BEQ      2$
3439              ;*
3440              ;*THIS SECTION OF CODE ADDED TO MAKE SURE
3441              ;*THAT MASTER CLEAR, CLEARS THE MAR
3442              ;*
3443 015520 000240      NOP
3444 015522 000240      NOP
3445 015524 005011      CLR      @R1          ; CLEAR RUN, IF SET
3446 015526 052711 040000      BIS      #40000,(R1) ; SET MASTER CLEAR
3447 015532 005011      CLR      (R1)        ; CLEAR MASTER CLEAR
3448 015534      SROMCLK ; WE MUST FIRST CLOCK
3449 015540 121325      121325 ; THE MAR LATCH REGS
3450 015542      SROMCLK ; BEFORE WE CAN READ THEM
3451 015546 121304      121304
3452 015550      ROMCLK ; READ IBUS* <15> PUT IN PORT5
3453 015554 121325      121325 ; MAR HIGH
3454 015556      ROMCLK ; READ IBUS* <14>, PUT IN PORT4
3455 015562 121304      121304 ; MAR LOW
3456 015564 005002      CLR      R2          ; EXPECT MAR CLEAR
3457 015566 016104 000004      MOV      4(R1),R4    ; READ PORTS 4&5. THEY CONTAIN
3458              ; THE CONTENTS OF THE MAR
3459              ; MASTER CLEAR SHOULD HAVE
3460              ; CLEARED THE MAR
3461              ; BRANCH END TST IF CLEAR
3462 015572 042704 160000      BIC      #160000,R4  ; MASK OUT JUNK
3463 015576 020204      CMP      R2,R4      ; CHECK FOR MAR=0
3464 015600 001406      BEQ      40$
3465 015602              ERROR     44
3466 015606 104455      TRAP     C$ERDF
3467 015610 000054      .WORD   44
3468 015612 005425      .WORD   EMO
3469 015614 012176      .WORD   ERR44
3470 015616      40$:
3471 015616
3472              CHKREG
3473 015616 004737 003014      .LIST   ME
3474 015622 012702 015634      JSR     PC,RDKM     ; READ KMC11 REGISTERS
3475 015626 004737 003120      MOV     #64$,R2    ; LOAD POINTER
3476 015632 000405      JSR     PC,R$CHECK ; CHECK FOR NO CHANGE
3477 015634      000      000      000      64$: BR      65$
3478 015637      000      000      000      .BYTE  .....20

```

3479	015642	000	000	000
3480	015645	020		
3481	015646			
3482	015646			
3483	015646			
3484	015646	104401		

658:
ENDTST
L10052:
TRAP C\$ETST

```

3485 015650          BADHEAD
3486                :***** TEST 10 *****
3487                :*IOP (CRAM) ODT BITS TEST
3488                :*LOAD MAR WITH A 0 INC MAR UNTIL IT OVERFLOWS
3489                :*VERIFY THAT IBUS* 10 BITS IS SET ONLY WHEN MAR BIT 8 IS A ONE
3490                :*AND THAT IBUS* 10 BIT6 IS SET ON MAR OVERFLOW
3491 015650          BADHEAD
3492                :***** TEST 10 *****
3493
3494 015650          BGNTST
3495 015650          T10::
3496 015650
3497
3498 015654          MYINT
3499 015660 005002   MSTCLR
3500 015662 005002   CLR R2
3501 015666 010000   ROMCLK
3502 015670          1$: 010000
3503 015670          ROMCLK
3504 015674 121204   121204
3505 015676 005005   CLR R5
3506 015700 032702 000400 BIT #BIT8,R2
3507 015704 001402   BEQ 11$
3508 015706 012705 000040 MOV #BIT5,R5
3509 015712 016104 000004 11$: MOV 4(R1),R4
3510 015716 042704 177637 BIC #177637,R4
3511 015722 020504   CMP R5,R4
3512 015724 001410   BEQ 15$
3513 015726          ERROR 7
3514 015732 104455   TRAP C$ERDF
3515 015734 000007   .WORD 7
3516 015736 005425   .WORD EMO
3517 015740 007510   .WORD ERR7
3518 015742          ESCAPE TST
3519 015742 104410   TRAP C$ESCAPE
3520 015744 000166   .WORD L10053-.
3521 015746          15$:
3522 015746          ROMCLK
3523 015752 014000   014000
3524 015754 005202   INC R2
3525 015756 020237 002400 CMP R2,MEMSZ
3526 015762 001342   BNE 1$
3527 015764          ROMCLK
3528 015770 121204   121204
3529 015772 012705 000140 MOV #140,R5
3530 015776 016104 000004 MOV 4(R1),R4
3531 016002 042704 177627 BIC #177627,R4
3532 016006 020504   CMP R5,R4
3533 016010 001406   BEQ 17$
3534 016012          ERROR 7
3535 016016 104455   TRAP C$ERDF
3536 016020 000007   .WORD 7
3537 016022 005425   .WORD EMO
3538 016024 007510   .WORD ERR7
3539 016026          17$:
3540 016032 010000   ROMCLK
                                010000
                                :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
                                :MAR_0
  
```


3541	016034					ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3542	016040	004000				004000		:MAR HI 0
3543	016042					ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3544	016046	121204				121204		:PORT4 IBUS* 10
3545	016050	005005				CLR	R5	:R5='EXPECTED'
3546	016052	016104	000004			MOV	4(R1),R4	:R4='FOUND'
3547	016056	042704	177637			BIC	#177637,R4	:CLEAR UNWANTED BITS
3548	016062	020504				CMP	R5,R4	:BITS 5&6 SHOULD BE CLEAR
3549	016064	001406				BEQ	2\$:BR IF OK
3550	016066					ERROR	7	:ERROR 5&6 NOT BOTH CLEAR
3551	016072	104455				TRAP	C\$ERDF	
3552	016074	000007				.WORD	7	
3553	016076	005425				.WORD	EMO	
3554	016100	007510				.WORD	ERR7	
3555	016102				2\$:			
3556	016102					CHKREG		
3557						.LIST	ME	
3558	016102	004737	003014			JSR	PC,RDKM	: READ KMC11 REGISTERS
3559	016106	012702	016120			MOV	#64\$,R2	: LOAD POINTER
3560	016112	004737	003120			JSR	PC,RCHECK	: CHECK FOR NO CHANGE
3561	016116	000405				BR	65\$: SKIP LIST
3562	016120	000	000	000	64\$:	.BYTE20	
3563	016123	000	000	000				
3564	016126	000	000	000				
3565	016131	020						
3566	016132				65\$:			
3567	016132				ENDTST			
3568	016132				L10053:			
3569	016132	104401				TRAP	C\$ETST	

```
3570 016134 BADHEAD
3571 :***** TEST 11 *****
3572 :*CRAM TEST OF JUMP(1) NEVER MICRO-PROCESSOR INSTRUCTION.
3573 :*PERFORM THE JUMP INSTRUCTION
3574 :*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
3575 :*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
3576 :*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
3577 :*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
3578 :*THE CRAM PC IS CORRECT. IF THE CRAM PC IS NOT RIGHT,
3579 :*THEN PORT4 CONTAINS A 37
3580 016134 BADHEAD
3581 :***** TEST 11 *****
3582
3583 BGNTST
3584 T11::
3585 10$:
3586 016134
3587
3588 016140 MYINT
3589 016144 MSTCLR
3590 016144 104404 TRAP C$BSEG ;R1 CONTAINS BASE M8200,4,6,7 ADDRESS
3591 016146 004737 004222 JSR PC, MEMSET ;MASTER CLEAR M8200,4,6,7
3592 016152 1$: JSR PC, CLRALL ;SET MEM AND RAM
3593 016152 004737 003750 SR0MCLK ;CLEAR ALL CONDITIONS
3594 016156 100400 100400 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3595 016162 115377 004112 SR0MCLK ;START AT ROM PC=0
3596 016170 004737 004112 114377!<400*2> ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3597 016172 000001 JSR PC, RAMDAT ;JUMP TO ROM PC OF 1777
3598 016176 020504 1 ;R4=CRAM PC (LSB 8 BITS)
3599 016200 001406 1 ;EXPECTED DATA
3600 016202 104455 004112 CMP R5,R4 ;IS ROM PC CORRECT?
3601 016204 000005 BEQ 2$ ;BR IF NO
3602 016210 000005 ERROR 5 ;ERROR, CRAM PC IS WRONG
3603 016212 005425 TRAP C$ERDF
3604 016214 007160 .WORD 5
3605 016216 104410 .WORD EMO
3606 016220 000002 2$: ESCAPE SEG ;ERR5
3607 016222 000002 TRAP C$ESCAPE
3608 016224 104405 10000$: .WORD 10000$-.
3609 016224 104405 TRAP C$ESEG
3610 016224 104405 BGNSEG
3611 016226 104404 TRAP C$BSEG
3612 016230 000240 NOP
3613 016232 000240 NOP
3614 016234 004737 003750 JSR PC, CLRALL ;CLEAR ALL CONDITIONS
3615 016240 100403 SR0MCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3616 016244 101000 100403 ;START AT ROM PC=3
3617 016252 004737 004112 SR0MCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3618 016254 000001 100000!<400*2> ;JUMP TO ROM PC OF 0
3619 016260 020504 004112 JSR PC, RAMDAT ;R4=CRAM PC (LSB 8 BITS)
3620 016262 000001 1 ;EXPECTED DATA
3621 016264 001406 001406 CMP R5,R4 ;IS ROM PC CORRECT?
3622 BEQ 4$ ;BR IF YES
```

3626	016266					ERROR	5		:ERROR, CROM PC IS WRONG
3627	016272	104455				TRAP	C\$ERDF		
3628	016274	000005				.WORD	5		
3629	016276	005425				.WORD	EMO		
3630	016300	007160				.WORD	ERR5		
3631	016302				4\$:	ESCAPE	SEG		
3632	016302	104410				TRAP	C\$ESCAPE		
3633	016304	000002				.WORD	10001\$-		
3634	016306					ENDSEG			
3635	016306				10001\$:				
3636	016306	104405				TRAP	C\$ESEG		
3637	016310					BGNSEG			
3638	016310	104404				TRAP	C\$BSEG		
3639	016312	004737	003750			JSR	PC,CLRALL		:CLEAR ALL CONDITINS
3640	016316					SROMCLK			:NEXT WORD IS INSTRUCION, ROMCLK PC=5304
3641	016322	100406				100406			:START AT ROM PC=6
3642	016324					SROMCLK			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3643	016330	105125				104125!<400*2>			:JUMP TO ROM PC OF 525
3644	016332	004737	004112			JSR	PC,RAMDAT		:R4=CROM PC (LSB 8 BITS)
3645	016336	000001				1			:EXPECTED DATA
3646	016340	020504				CMP	R5,R4		:IS ROM PC CORRECT?
3647	016342	001406				BEQ	6\$:BR IF YES
3648	016344					ERROR	5		:ERROR, CROM PC IS WRONG
3649	016350	104455				TRAP	C\$ERDF		
3650	016352	000005				.WORD	5		
3651	016354	005425				.WORD	EMO		
3652	016356	007160				.WORD	ERR5		
3653	016360				6\$:	ESCAPE	SEG		
3654	016360	104410				TRAP	C\$ESCAPE		
3655	016362	000002				.WORD	10002\$-		
3656	016364					ENDSEG			
3657	016364				10002\$:				
3658	016364	104405				TRAP	C\$ESEG		
3659	016366					CHKREG			
3660						.LIST	ME		
3661	016366	004737	003014			JSR	PC,RDKM		: READ KMC11 REGISTERS
3662	016372	012702	016404			MOV	#64\$,R2		: LOAD POINTER
3663	016376	004737	003120			JSR	PC,RCHECK		: CHECK FOR NO CHANGE
3664	016402	000405				BR	65\$: SKIP LIST
3665	016404	000	000	000	64\$:	.BYTE20		
3666	016407	000	000	000					
3667	016412	000	000	000					
3668	016415	020							
3669	016416				65\$:				
3670	016416				ENDTST				
3671	016416				L10054:				
3672	016416	104401				TRAP	C\$ETST		


```
3673 016420 BADHEAD
3674 :***** TEST 12 *****
3675 :*CRAM TEST OF JUMP(I) ALWAYS MICRO-PROCESSOR INSTRUCTION.
3676 :*PERFORM THE JUMP INSTRUCTION
3677 :*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
3678 :*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
3679 :*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
3680 :*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
3681 :*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
3682 :*THEN PORT4 WILL CONTAIN A 37
3683 016420 BADHEAD
3684 :***** TEST 12 *****
3685
3686 016420 BGNTST
3687 016420 T12::
3688 016420
3689 MYINT
3690 016424 :R1 CONTAINS BASE M8200,4,6,7 ADDRESS
3691 016430 004737 004222 MSTCLR :MASTER CLEAR M8200,4,6,7
3692 016434 1$: JSR PC, MEMSET :SET MEM AND RAM
3693 016434 104404 TRAP C$BSEG
3694 016436 SR0MCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3695 016442 100400 100400 :START AT ROM PC=0
3696 016444 SR0MCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3697 016450 114777 114377!<400*1> :JUMP TO ROM PC OF 1777
3698 016452 004737 004112 JSR PC, RAMDAT :R4=CRAM PC (LSB 8 BITS)
3699 016456 000377 377 :EXPECTED DATA
3700 016460 120504 CMPB R5,R4 :IS ROM PC CORRECT?
3701 016462 001406 BEQ 2$ :BR IF YES
3702 016464 ERROR 5 :ERROR, CRAM PC IS WRONG
3703 016470 104455 TRAP C$ERDF
3704 016472 000005 .WORD 5
3705 016474 005425 .WORD EMO
3706 016476 007160 .WORD ERR5
3707 016500 2$: ESCAPE SEG
3708 016500 104410 TRAP C$ESCAPE
3709 016502 000002 .WORD 10000$-.
3710 016504 ENDSEG
3711 016504 10000$: TRAP C$ESEG
3712 016504 104405 BGNSEG
3713 016506 TRAP C$BSEG
3714 016506 104404 SR0MCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3715 016510 100403 100403 :START AT ROM PC=3
3716 016514 100403 SR0MCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3717 016516 100400 100000!<400*1> :JUMP TO ROM PC OF 0
3718 016522 100400 004737 004112 JSR PC, RAMDAT :R4=CRAM PC (LSB 8 BITS)
3719 016524 004737 000000 0 :EXPECTED DATA
3720 016530 000000 120504 CMPB R5,R4 :IS ROM PC CORRECT?
3721 016532 120504 001406 BEQ 4$ :BR IF YES
3722 016534 001406 ERROR 5 :ERROR, CRAM PC IS WRONG
3723 016536 104455 TRAP C$ERDF
3724 016542 104455 .WORD 5
3725 016544 000005 .WORD EMO
3726 016546 005425 .WORD ERR5
3727 016550 007160 4$: ESCAPE SEG
3728 016552
```

```
3729 016552 104410 TRAP C$ESCAPE
3730 016554 000002 .WORD 10001$-.
3731 016556 ENDSEG
3732 016556 10001$:
3733 016556 104405 TRAP C$ESEG
3734 016560 BGNSEG
3735 016560 104404 TRAP C$BSEG
3736 016562 SRMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3737 016566 100406 100406 ;START AT ROM PC=6
3738 016570 SRMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3739 016574 104525 104125!<400*1> ;JUMP TO ROM PC OF 525
3740 016576 004737 004112 JSR PC,RAMDAT ;R4=CRAM PC (LSB 8 BITS)
3741 016602 000125 125 ;EXPECTED DATA
3742 016604 120504 CMPB R5,R4 ;IS ROM PC CORRECT?
3743 016606 001406 BEQ 6$ ;BR IF YES
3744 016610 ERROR 5 ;ERROR, CRAM PC IS WRONG
3745 016614 104455 TRAP C$ERDF
3746 016616 000005 .WORD 5
3747 016620 005425 .WORD EMO
3748 016622 007160 .WORD ERR5
3749 016624 6$: ESCAPE SEG
3750 016624 104410 TRAP C$ESCAPE
3751 016626 000002 .WORD 10002$-.
3752 016630 ENDSEG
3753 016630 10002$:
3754 016630 104405 TRAP C$ESEG
3755 016632 CHKREG
3756 .LIST ME
3757 016632 004737 003014 JSR PC,RDKM ; READ KMC11 REGISTERS
3758 016636 012702 016650 MOV #64$,R2 ; LOAD POINTER
3759 016642 004737 003120 JSR PC,RCHECK ; CHECK FOR NO CHANGE
3760 016646 000405 BR 65$ ; SKIP LIST
3761 016650 000 000 000 64$: .BYTE .....20
3762 016653 000 000 000
3763 016656 000 000 000
3764 016661 020
3765 016662 65$:
3766 016662 ENDTST
3767 016662 L10055:
3768 016662 104401 TRAP C$ETST
```

```
3769 016664 BADHEAD
3770 :***** TEST 13 *****
3771 :*CRAM TEST OF JUMP(I) ON C BIT SET MICRO-PROCESSOR INSTRUCTION.
3772 :*SET THE C BIT, PERFORM THE JUMP INSTRUCTION.
3773 :*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
3774 :*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
3775 :*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
3776 :*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
3777 :*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
3778 :*THEN PORT4 WILL CONTAIN A 37
3779 016664 BADHEAD
3780 :***** TEST 13 *****
3781
3782 016664 BGNTST
3783 016664 T13::
3784 016664
3785 MYINT
3786 016670 MSTCLR ;R1 CONTAINS BASE M8200,4,6,7 ADDRESS
3787 016674 004737 004222 JSR PC, MEMSET ;MASTER CLEAR M8200,4,6,7
3788 016700 BGNSEG ;SET MEM AND RAM
3789 016700 104404 TRAP C$BSEG
3790 016702 004737 004042 JSR PC, SETC ;SET THE C BIT'
3791 016706 SRMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3792 016712 100400 100400 ;START AT ROM PC=0
3793 016714 SRMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3794 016720 115377 114377!<400*2> ;JUMP TO ROM PC OF 1777
3795 016722 004737 004112 JSR PC, RAMDAT ;R4=CRAM PC (LSB 8 BITS)
3796 016726 000377 377 ;EXPECTED DATA
3797 016730 120504 CMPB R5, R4 ;IS ROM PC CORRECT?
3798 016732 001406 BEQ 2$ ;BR IF YES
3799 016734 ERROR 5 ;ERROR, CRAM PC IS WRONG
3800 016740 104455 TRAP C$ERDF
3801 016742 000005 .WORD 5
3802 016744 005425 .WORD EMO
3803 016746 007160 .WORD ERR5
3804 016750 2$: ;LOOP TO 1$ IF SW09=1
3805 016750 ESCAPE SEG
3806 016750 104410 TRAP C$ESCAPE
3807 016752 000002 .WORD 10000$-.
3808 016754 ENDSEG
3809 016754 10000$:
3810 016754 104405 TRAP C$ESEG
3811 016756 BGNSEG
3812 016756 104404 TRAP C$BSEG
3813 016760 004737 004042 JSR PC, SETC ;SET THE C BIT'
3814 016764 SRMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3815 016770 100403 100403 ;START AT ROM PC=3
3816 016772 SRMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3817 016776 101000 100000!<400*2> ;JUMP TO ROM PC OF 0
3818 017000 004737 004112 JSR PC, RAMDAT ;R4=CRAM PC (LSB 8 BITS)
3819 017004 000000 0 ;EXPECTED DATA
3820 017006 120504 CMPB R5, R4 ;IS ROM PC CORRECT?
3821 017010 001406 BEQ 4$ ;BR IF YES
3822 017012 ERROR 5 ;ERROR, CRAM PC IS WRONG
3823 017016 104455 TRAP C$ERDF
3824 017020 000005 .WORD 5
```



```
3825 017022 005425          .WORD  EMO
3826 017024 007160          .WORD  ERR5
3827 017026                4$:          ;LOOP TO 3$ IF SW09=1
3828 017026                ESCAPE SEG
3829 017026 104410          TRAP   C$ESCAPE
3830 017030 000002          .WORD  10001$-.
3831 017032                ENDSEG
3832 017032                10001$:
3833 017032 104405          TRAP   C$ESEG
3834 017034                BGNSEG
3835 017034 104404          TRAP   C$BSEG
3836 017036 004737 004042    JSR    PC,SETC          ;SET THE C BIT'
3837 017042                SRMCLK          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3838 017046 100406          100406          ;START AT ROM PC=6
3839 017050                SRMCLK          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3840 017054 105125          104125!<400*2>      ;JUMP TO ROM PC OF 525
3841 017056 004737 004112    JSR    PC,RAMDAT      ;R4=CRAM PC (LSB 8 BITS)
3842 017062 000125          125              ;EXPECTED DATA
3843 017064 120504          CMPB   R5,R4        ;IS ROM PC CORRECT?
3844 017066 001406          BEQ   6$           ;BR IF YES
3845 017070                ERROR          5              ;ERROR, CRAM PC IS WRONG
3846 017074 104455          TRAP   C$ERDF
3847 017076 000005          .WORD  5
3848 017100 005425          .WORD  EMO
3849 017102 007160          .WORD  ERR5
3850 017104                6$:          ESCAPE SEG
3851 017104 104410          TRAP   C$ESCAPE
3852 017106 000002          .WORD  10002$-.
3853 017110                ENDSEG
3854 017110                10002$:
3855 017110 104405          TRAP   C$ESEG
3856 017112                CHKREG
3857 017112                .LIST  ME
3858 017112 004737 003014      JSR    PC,RDKM        ; READ KMC11 REGISTERS
3859 017116 012702 017130      MOV   #64$,R2        ; LOAD POINTER
3860 017122 004737 003120      JSR    PC,RCHECK      ; CHECK FOR NO CHANGE
3861 017126 000405                BR    65$           ; SKIP LIST
3862 017130                000          000          000 64$: .BYTE .....20
3863 017133                000          000          000
3864 017136                000          000          000
3865 017141 020
3866 017142                65$:
3867 017142                ENDTST
3868 017142                L10056:
3869 017142 104401          TRAP   C$ETST
```

```
3870 017144 BADHEAD
3871 :***** TEST 14 *****
3872 :*CRAM TEST OF JUMP(I) ON Z BIT SET MICRO-PROCESSOR INSTRUCTION.
3873 :*SET THE Z BIT, PERFORM THE JUMP INSTRUCTION.
3874 :*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
3875 :*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
3876 :*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
3877 :*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
3878 :*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
3879 :*THEN PORT4 WILL CONTAIN A 37
3880 017144 BADHEAD
3881 :***** TEST 14 *****
3882
3883 017144 BGNTST
3884 017144 T14::
3885 017144
3886 MYINT
3887 017150 MSTCLR ;R1 CONTAINS BASE M8200,4,6,7 ADDRESS
3888 017154 004737 004222 JSR PC,MEMSET ;MASTER CLEAR M8200,4,6,7
3889 017160 1$: BGNSEG ;SET MEM AND RAM
3890 017160 104404 TRAP C$BSEG
3891 017162 004737 004074 JSR PC,SETZ ;SET THE Z BIT'
3892 017166 SROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3893 017172 100400 100400 ;START AT ROM PC=0
3894 017174 SROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3895 017200 115777 114377!<400*3> ;JUMP TO ROM PC OF 1777
3896 017202 004737 004112 JSR PC,RAMDAT ;R4=CRAM PC (LSB 8 BITS)
3897 017206 000377 377 ;EXPECTED DATA
3898 017210 120504 CMPB R5,R4 ;IS ROM PC CORRECT?
3899 017212 001406 BEQ 2$ ;BR IF YES
3900 017214 ERROR 5 ;ERROR, CRAM PC IS WRONG
3901 017220 104455 TRAP C$ERDF
3902 017222 000005 .WORD 5
3903 017224 005425 .WORD EMO
3904 017226 007160 .WORD ERR5
3905 017230 2$: ESCAPE SEG
3906 017230 104410 TRAP C$ESCAPE
3907 017232 000002 .WORD 10000$-.
3908 017234 ENDSEG
3909 017234 10000$: TRAP C$ESEG
3910 017234 104405 BGNSEG
3911 017236 TRAP C$BSEG
3912 017236 104404 JSR PC,SETZ ;SET THE Z BIT'
3913 017240 004737 004074 SROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3914 017244 100403 100403 ;START AT ROM PC=3
3915 017250 SROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3916 017252 101400 100000!<400*3> ;JUMP TO ROM PC OF 0
3917 017256 004737 004112 JSR PC,RAMDAT ;R4=CRAM PC (LSB 8 BITS)
3918 017260 000000 0 ;EXPECTED DATA
3919 017264 120504 CMPB R5,R4 ;IS ROM PC CORRECT?
3920 017270 001406 BEQ 4$ ;BR IF YES
3921 017272 ERROR 5 ;ERROR, CRAM PC IS WRONG
3922 017276 104455 TRAP C$ERDF
3923 017300 000005 .WORD 5
3924 017302 005425 .WORD EMO
```

```
3926 017304 007160
3927 017306
3928 017306 104410
3929 017310 000002
3930 017312
3931 017312
3932 017312 104405
3933 017314
3934 017314 104404
3935 017316 004737 004074
3936 017322
3937 017326 100406
3938 017330
3939 017334 105525
3940 017336 004737 004112
3941 017342 000125
3942 017344 120504
3943 017346 001406
3944 017350
3945 017354 104455
3946 017356 000005
3947 017360 005425
3948 017362 007160
3949 017364
3950 017364 104410
3951 017366 000002
3952 017370
3953 017370
3954 017370 104405
3955 017372
3956
3957 017372 004737 003014
3958 017376 012702 017410
3959 017402 004737 003120
3960 017406 000405
3961 017410 000 000 000 64$:
3962 017413 000 000 000
3963 017416 000 000 000
3964 017421 020
3965 017422
3966 017422
3967 017422
3968 017422 104401

. WORD ERR5
4$: ESCAPE SEG
TRAP C$ESCAPE
. WORD 10001$-.
ENDSEG
10001$:
TRAP C$ESEG
BGNSEG
TRAP C$BSEG
JSR PC,SETZ ;SET THE Z BIT'
SROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
100406 ;START AT ROM PC=6
SROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
104125!<400*3> ;JUMP TO ROM PC OF 525
JSR PC,RAMDAT ;R4=CRAM PC (LSB 8 BITS)
125 ;EXPECTED DATA
CMPB R5,R4 ;IS ROM PC CORRECT?
BEQ 6$ ;BR IF YES
ERROR 5 ;ERROR, CRAM PC IS WRONG
TRAP C$ERDF
. WORD 5
. WORD EMO
. WORD ERR5
6$: ESCAPE SEG
TRAP C$ESCAPE
. WORD 10002$-.
ENDSEG
10002$:
TRAP C$ESEG
CHKREG
. LIST ME
JSR PC,RDKM ; READ KMC11 REGISTERS
MOV #64$,R2 ; LOAD POINTER
JSR PC,RCHECK ; CHECK FOR NO CHANGE
BR 65$ ; SKIP LIST
. BYTE .....20
65$:
ENDTST
L10057:
TRAP C$ETST
```



```
3969 017424 BADHEAD
3970 :***** TEST 15 *****
3971 :*CRAM TEST OF JUMP(I) ON BRO SET MICRO-PROCESSOR INSTRUCTION.
3972 :*SET THE BRO BIT, PERFORM THE JUMP INSTRUCTION.
3973 :*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
3974 :*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
3975 :*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
3976 :*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
3977 :*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
3978 :*THEN PORT4 WILL CONTAIN A 37
3979 017424 BADHEAD
3980 :***** TEST 15 *****
3981
3982 017424 BGNTST
3983 017424 T15::
3984 017424
3985
3986 017430 MYINT
3987 017434 004737 004222 MSTCLR
3988 017440 JSR PC, MEMSET
3989 017440
3990 017440 104404 BGNSEG
3991 017442 004737 004002 TRAP CSBSEG
3992 017446 SRMCLK PC, SETBRO
3993 017452 100400 100400 :SET THE BRO BIT'
3994 017454 SRMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3995 017460 116377 114377! <400*4> :START AT ROM PC=0
3996 017462 004737 004112 JSR PC, RAMDAT :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3997 017466 000377 377 :JUMP TO ROM PC OF 1777
3998 017470 120504 CMPB R5, R4 :R4=CRAM PC (LSB 8 BITS)
3999 017472 001406 BEQ 2$ :EXPECTED DATA
4000 017474 ERROR 5 :IS ROM PC CORRECT?
4001 017500 104455 TRAP CSERDF :BR IF YES
4002 017502 000005 .WORD 5 :ERROR, CRAM PC IS WRONG
4003 017504 005425 .WORD EMO
4004 017506 007160 .WORD ERR5
4005 017510 2$: ESCAPE SEG
4006 017510 104410 TRAP C$ESCAPE
4007 017512 000002 .WORD 10000$-.
4008 017514 ENDSEG
4009 017514 10000$:
4010 017514 104405 TRAP C$ESEG
4011 017516 BGNSEG
4012 017516 104404 TRAP CSBSEG
4013 017520 004737 004002 JSR PC, SETBRO
4014 017524 SRMCLK :SET THE BRO BIT'
4015 017530 100403 100403 :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4016 017532 SRMCLK :START AT ROM PC=3
4017 017536 102000 100000! <400*4> :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4018 017540 004737 004112 JSR PC, RAMDAT :JUMP TO ROM PC OF 0
4019 017544 000000 0 :R4=CRAM PC (LSB 8 BITS)
4020 017546 120504 CMPB R5, R4 :EXPECTED DATA
4021 017550 001406 BEQ 4$ :IS ROM PC CORRECT?
4022 017552 ERROR 5 :BR IF YES
4023 017556 104455 TRAP CSERDF :ERROR, CRAM PC IS WRONG
4024 017560 000005 .WORD 5
```

4025 017562 005425
4026 017564 007160
4027 017566
4028 017566 104410
4029 017570 000002
4030 017572
4031 017572
4032 017572 104405
4033 017574
4034 017574 104404
4035 017576 004737 004002
4036 017602
4037 017606 100406
4038 017610
4039 017614 106125
4040 017616 004737 004112
4041 017622 000125
4042 017624 120504
4043 017626 001406
4044 017630
4045 017634 104455
4046 017636 000005
4047 017640 005425
4048 017642 007160
4049 017644
4050 017644 104410
4051 017646 000002
4052 017650
4053 017650
4054 017650 104405
4055 017652
4056
4057 017652 004737 003014
4058 017656 012702 017670
4059 017662 004737 003120
4060 017666 000405
4061 017670 000 000 000
4062 017673 000 000 000
4063 017676 000 000 000
4064 017701 020
4065 017702
4066 017702
4067 017702
4068 017702 104401

4\$: .WORD EMO
.WORD ERR5
ESCAPE SEG
TRAP C\$ESCAPE
.WORD 10001\$-
ENDSEG
10001\$: TRAP C\$ESEG
BGNSEG
TRAP C\$BSEG
JSR PC,SETBRO
SROMCLK
100406
SROMCLK
104125! <400*4>
JSR PC,RAMDAT
125
CMPB R5,R4
BEQ 6\$
ERROR 5
TRAP C\$ERDF
.WORD 5
.WORD EMO
.WORD ERR5
6\$: ESCAPE SEG
TRAP C\$ESCAPE
.WORD 10002\$-
ENDSEG
10002\$: TRAP C\$ESEG
CHKREG
.LIST ME
JSR PC,RDKM
MOV #64\$,R2
JSR PC,RCHECK
BR 65\$
.BYTE20
65\$:
ENDTST
L10060:
TRAP C\$ETST

:SET THE BRO BIT'
:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
:START AT ROM PC=6
:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
:JUMP TO ROM PC OF 525
:R4=CRAM PC (LSB 8 BITS)
:EXPECTED DATA
:IS ROM PC CORRECT?
:BR IF YES
:ERROR, CRAM PC IS WRONG

: READ KMC11 REGISTERS
: LOAD POINTER
: CHECK FOR NO CHANGE
: SKIP LIST

```

4069 017704      BADHEAD
4070             :***** TEST 16 *****
4071             :*CRAM TEST OF JUMP(I) ON BR1 SET MICRO-PROCESSOR INSTRUCTION.
4072             :*SET THE BR1 BIT, PERFORM THE JUMP INSTRUCTION.
4073             :*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
4074             :*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4075             :*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4076             :*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4077             :*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
4078             :*THEN PORT4 WILL CONTAIN A 37
4079 017704      BADHEAD
4080             :***** TEST 16 *****
4081
4082 017704      BGNTST
4083 017704      T16::
4084 017704
4085
4086 017710      MYINT
4087 017714      004737 004222      MSTCLR
4088 017720      1$:          JSR      PC, MEMSET      ;R1 CONTAINS BASE M8200,4,6,7 ADDRESS
4089 017720      BGNSEG      ;MASTER CLEAR M8200,4,6,7
4090 017720      104404      TRAP     CSBSEG      ;SET MEM AND RAM
4091 017722      004737 004012      JSR      PC, SETBR1
4092 017726      SROMCLK
4093 017732      100400      100400      ;SET THE BR1 BIT'
4094 017734      SROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4095 017740      116777      114377! <400*5> ;START AT ROM PC=0
4096 017742      004737 004112      JSR      PC, RAMDAT ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4097 017746      000377      377          ;JUMP TO ROM PC OF 1777
4098 017750      120504      CMPB     R5, R4      ;R4=CRAM PC (LSB 8 BITS)
4099 017752      001406      BEQ      2$          ;EXPECTED DATA
4100 017754      ERROR      5          ;IS ROM PC CORRECT?
4101 017760      104455      TRAP     C$ERDF     ;BR IF YES
4102 017762      000005      .WORD   5          ;ERROR, CRAM PC IS WRONG
4103 017764      005425      .WORD   EMO
4104 017766      007160      .WORD   ERR5
4105 017770      2$:      ESCAPE  SEG
4106 017770      104410      TRAP     C$ESCAPE
4107 017772      000002      .WORD   10000$-.
4108 017774      ENDSEG
4109 017774      10000$:
4110 017774      104405      TRAP     C$ESEG
4111 017776      BGNSEG
4112 017776      104404      TRAP     CSBSEG
4113 020000      004737 004012      JSR      PC, SETBR1      ;SET THE BR1 BIT'
4114 020004      SROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4115 020010      100403      100403      ;START AT ROM PC=3
4116 020012      SROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4117 020016      102400      100000! <400*5> ;JUMP TO ROM PC OF 0
4118 020020      004737 004112      JSR      PC, RAMDAT      ;R4=CRAM PC (LSB 8 BITS)
4119 020024      000000      0          ;EXPECTED DATA
4120 020026      120504      CMPB     R5, R4      ;IS ROM PC CORRECT?
4121 020030      001406      BEQ      4$          ;BR IF YES
4122 020032      ERROR      5          ;ERROR, CRAM PC IS WRONG
4123 020036      104455      TRAP     C$ERDF
4124 020040      000005      .WORD   5

```



```

4169 020164      BADHEAD
4170             :***** TEST 17 *****
4171             :*CRAM TEST OF JUMP(1) ON BR4 SET MICRO-PROCESSOR INSTRUCTION.
4172             :*SET THE BR4 BIT, PERFORM THE JUMP INSTRUCTION.
4173             :*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
4174             :*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4175             :*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4176             :*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4177             :*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
4178             :*THEN PORT4 WILL CONTAIN A 37
4179 020164      BADHEAD
4180             :***** TEST 17 *****
4181
4182 020164      BGNTST
4183 020164      T17::
4184 020164
4185 020170
4186 020174 004737 004222      1$: MYINT
4187 020200      MSTCLR           ;MASTER CLEAR M8200,4,6,7
4188 020200      JSR PC,MEMSET      ;SET MEM AND RAM
4189 020200 104404      BGNSEG
4190 020202 004737 004022      TRAP C$BSEG
4191 020206      JSR PC,SETBR4      ;SET THE BR4 BIT'
4192 020212 100400      SROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4193 020214      100400           ;START AT ROM PC=0
4194 020220 117377      SROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4195 020222 004737 004112      114377!<400*6>      ;JUMP TO ROM PC OF 1777
4196 020226 000377      JSR PC,RAMDAT ;R4=CRAM PC (LSB 8 BITS)
4197 020230 120504      377           ;EXPECTED DATA
4198 020232 001406      CMPB R5,R4      ;IS ROM PC CORRECT?
4199 020234      BEQ 2$           ;BR IF YES
4200 020240 104455      ERROR 5           ;ERROR, CRAM PC IS WRONG
4201 020242 000005      TRAP C$ERDF
4202 020244 005425      .WORD 5
4203 020246 007160      .WORD EMO
4204 020250      .WORD ERR5
4205 020250 104410      2$: ESCAPE SEG
4206 020252 000002      TRAP C$ESCAPE
4207 020254      .WORD 10000$-
4208 020254      ENDSEG
4209 020254 104405      10000$: TRAP C$ESEG
4210 020256      BGNSEG
4211 020256 104404      TRAP C$BSEG
4212 020260 004737 004022      JSR PC,SETBR4      ;SET THE BR4 BIT'
4213 020264      SROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4214 020270 100403      100403           ;START AT ROM PC=3
4215 020272      SROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4216 020276 103000      100000!<400*6>      ;JUMP TO ROM PC OF 0
4217 020300 004737 004112      JSR PC,RAMDAT ;R4=CRAM PC (LSB 8 BITS)
4218 020304 000000      0           ;EXPECTED DATA
4219 020306 120504      CMPB R5,R4      ;IS ROM PC CORRECT?
4220 020310 001406      BEQ 4$           ;BR IF YES
4221 020312      ERROR 5           ;ERROR, CRAM PC IS WRONG
4222 020316 104455      TRAP C$ERDF
4223 020320 000005      .WORD 5
4224 020322 005425      .WORD EMO

```



```
4268 020444 BADHEAD
4269 :***** TEST 18 *****
4270 :*CRAM TEST OF JUMP(I) ON BR7 SET MICRO-PROCESSOR INSTRUCTION.
4271 :*SET THE BR7 BIT, PERFORM THE JUMP INSTRUCTION.
4272 :*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
4273 :*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4274 :*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4275 :*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4276 :*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
4277 :*THEN PORT4 WILL CONTAIN A 37
4278 020444 BADHEAD
4279 :***** TEST 18 *****
4280
4281 020444 BGNTST
4282 020444 T18::
4283 020444
4284 MYINT
4285 020450 :R1 CONTAINS BASE M8200,4,6,7 ADDRESS
4286 020454 004737 004222 MSTCLR :MASTER CLEAR M8200,4,6,7
4287 020460 JSR PC, MEMSET :SET MEM AND RAM
4288 020460 104404 1$: BGNSEG
4289 020462 004737 004032 TRAP CSBSEG
4290 020466 JSR PC, SETBR7 :SET THE BR7 BIT'
4291 020472 100400 SR0MCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4292 020474 100400 :START AT ROM PC=0
4293 020500 117777 SR0MCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4294 020502 004737 004112 114377!<400*7> JSR PC, RAMDAT :JUMP TO ROM PC OF 1777
4295 020506 000377 377 :R4=CRAM PC (LSB 8 BITS)
4296 020510 120504 CMPB R5,R4 :EXPECTED DATA
4297 020512 001406 BEQ 2$ :IS ROM PC CORRECT?
4298 020514 ERROR 5 :BR IF YES
4299 020520 104455 TRAP C$ERDF :ERROR, CRAM PC IS WRONG
4300 020522 000005 .WORD 5
4301 020524 005425 .WORD EMO
4302 020526 007160 .WORD ERR5
4303 020530 2$: ESCAPE SEG
4304 020530 104410 TRAP C$ESCAPE
4305 020532 000002 .WORD 10000$-.
4306 020534 ENDSEG
4307 020534 10000$:
4308 020534 104405 TRAP C$ESEG
4309 020536 BGNSEG
4310 020536 104404 TRAP CSBSEG
4311 020540 004737 004032 JSR PC, SETBR7 :SET THE BR7 BIT'
4312 020544 SR0MCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4313 020550 100403 100403 :START AT ROM PC=3
4314 020552 SR0MCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4315 020556 103400 100000!<400*7> JSR PC, RAMDAT :JUMP TO ROM PC OF 0
4316 020560 004737 004112 0 :R4=CRAM PC (LSB 8 BITS)
4317 020564 000000 0 :EXPECTED DATA
4318 020566 120504 CMPB R5,R4 :IS ROM PC CORRECT?
4319 020570 001406 BEQ 4$ :BR IF YES
4320 020572 ERROR 5 :ERROR, CRAM PC IS WRONG
4321 020576 104455 TRAP C$ERDF
4322 020600 000005 .WORD 5
4323 020602 005425 .WORD EMO
```

4324 020604 007160
4325 020606
4326 020606 104410
4327 020610 000002
4328 020612
4329 020612
4330 020612 104405
4331 020614
4332 020614 104404
4333 020616 004737 004032
4334 020622
4335 020626 100406
4336 020630
4337 020634 107525
4338 020636 004737 004112
4339 020642 000125
4340 020644 120504
4341 020646 001406
4342 020650
4343 020654 104455
4344 020656 000005
4345 020660 005425
4346 020662 007160
4347 020664
4348 020664 104410
4349 020666 000002
4350 020670
4351 020670
4352 020670 104405
4353 020672
4354
4355 020672 004737 003014
4356 020676 012702 020710
4357 020702 004737 003120
4358 020706 000405
4359 020710 000 000 000 000
4360 020713 000 000 000
4361 020716 000 000
4362 020721 020
4363 020722
4364 020722
4365 020722
4366 020722 104401

4\$: .WORD ERR5
ESCAPE SEG
TRAP C\$ESCAPE
.WORD 10001\$-.
ENDSEG
10001\$: TRAP C\$ESEG
BGNSEG
TRAP C\$BSEG
JSR PC,SETBR7
SROMCLK
100406
SROMCLK
104125!<400*7>
JSR PC,RAMDAT
125
CMPB R5,R4
BEQ 6\$
ERROR 5
TRAP C\$ERDF
.WORD 5
.WORD EMO
.WORD ERR5
6\$: ESCAPE SEG
TRAP C\$ESCAPE
.WORD 10002\$-.
ENDSEG
10002\$: TRAP C\$ESEG
CHKREG
.LIST ME
JSR PC,RDKM
MOV #64\$,R2
JSR PC,RCHECK
BR 65\$
64\$: .BYTE20
65\$:
ENDTST
L10063: TRAP C\$ETST

:SET THE BR7 BIT'
:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
:START AT ROM PC=6
:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
:JUMP TO ROM PC OF 525
:R4=CRAM PC (LSB 8 BITS)
:EXPECTED DATA
:IS ROM PC CORRECT?
:BR IF YES
:ERROR, CRAM PC IS WRONG

: READ KMC11 REGISTERS
: LOAD POINTER
: CHECK FOR NO CHANGE
: SKIP LIST

```
4367 020724 BADHEAD
4368 :***** TEST 19 *****
4369 :*CRAM TEST OF NO JUMP(I) ON C BIT CLEAR MICRO-PROCESSOR INSTRUCTION.
4370 :*CLEAR THE C BIT, PERFORM THE JUMP INSTRUCTION.
4371 :*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
4372 :*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4373 :*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4374 :*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4375 :*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
4376 :*THEN PORT4 WILL CONTAIN A 37
4377 020724 BADHEAD
4378 :***** TEST 19 *****
4379
4380 020724 BGNTST
4381 020724 T19::
4382 020724
4383 020730 MYINT
4384 020734 004737 004222 MSTCLR ;MASTER CLEAR M8200,4,6,7
4385 020740 1$: JSR PC, MEMSET ;SET MEM AND RAM
4386 020740 104404 BGNSEG
4387 020742 TRAP C$BSEG
4388 020746 MSTCLR
4389 020752 100400 SR0MCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4390 020754 100400 ;START AT ROM PC=0
4391 020760 115377 SR0MCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4392 020762 004737 004112 114377!<400*2> ;JUMP TO ROM PC OF 1777
4393 020766 000001 JSR PC, RAMDAT ;R4=CRAM PC (LSB 8 BITS)
4394 020770 020504 1 ;EXPECTED DATA
4395 020772 001406 CMP R5,R4 ;IS ROM PC CORRECT?
4396 020774 BEQ 2$ ;BR IF YES
4397 021000 104455 ERROR 5 ;ERROR, CRAM PC IS WRONG
4398 021002 000005 TRAP C$ERDF
4399 021004 005425 .WORD 5
4400 021006 007160 .WORD EMO
4401 021010 2$: ESCAPE SEG ;ERR5
4402 021010 104410 TRAP C$ESCAPE
4403 021012 000002 .WORD 10000$-.
4404 021014 ENDSEG
4405 021014 10000$: TRAP C$ESEG
4406 021014 104405 BGNSEG
4407 021016 104404 TRAP C$BSEG
4408 021020 000240 NOP
4409 021022 000240 NOP
4410 021024 004737 003750 JSR PC, CLRALL ;CLEAR ALL CONDITIONS
4411 021030 100403 SR0MCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4412 021034 100403 ;START AT ROM PC=3
4413 021036 101000 SR0MCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4414 021042 004737 004112 100000!<400*2> ;JUMP TO ROM PC OF 0
4415 021044 000001 JSR PC, RAMDAT ;R4=CRAM PC (LSB 8 BITS)
4416 021050 000001 1 ;EXPECTED DATA
4417 021052 020504 CMP R5,R4 ;IS ROM PC CORRECT?
4418 021054 001406 BEQ 4$ ;BR IF YES
4419 021056 104455 ERROR 5 ;ERROR, CRAM PC IS WRONG
4420 021062 104455 TRAP C$ERDF
4421 021064 000005 .WORD 5
```



```
4423 021066 005425      .WORD  EMO
4424 021070 007160      .WORD  ERR5
4425 021072           4$:  ESCAPE  SEG
4426 021072 104410      TRAP   C$ESCAPE
4427 021074 000002      .WORD  10001$-.
4428 021076           ENDSEG
4429 021076           10001$:
4430 021076 104405      TRAP   C$ESEG
4431 021100           BGNSEG
4432 021100 104404      TRAP   C$BSEG
4433 021102 004737 003750      JSR    PC,CLRALL
4434 021106           SROMCLK
4435 021112 100406      100406
4436 021114           SROMCLK
4437 021120 105125      104125!<400*2>
4438 021122 004737 004112      JSR    PC,RAMDAT
4439 021126 000001      1
4440 021130 020504      CMP    R5,R4
4441 021132 001406      BEQ   6$
4442 021134           ERROR  5
4443 021140 104455      TRAP   C$ERDF
4444 021142 000005      .WORD  5
4445 021144 005425      .WORD  EMO
4446 021146 007160      .WORD  ERR5
4447 021150           6$:  ESCAPE  SEG
4448 021150 104410      TRAP   C$ESCAPE
4449 021152 000002      .WORD  10002$-.
4450 021154           ENDSEG
4451 021154           10002$:
4452 021154 104405      TRAP   C$ESEG
4453 021156           CHKREG
4454           .LIST  ME
4455 021156 004737 003014      JSR    PC,RDKM
4456 021162 012702 021174      MOV   #64$,R2
4457 021166 004737 003120      JSR   PC,RCHECK
4458 021172 000405      BR    65$
4459 021174      000      000      000 64$: .BYTE .....20
4460 021177      000      000      000
4461 021202      000      000      000
4462 021205      020
4463 021206           65$:
4464 021206           ENDTST
4465 021206           L10064:
4466 021206 104401      TRAP   C$ETST

;CLEAR ALL CONDITIONS
;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
;START AT ROM PC=6
;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
;JUMP TO ROM PC OF 525
;R4=CRAM PC (LSB 8 BITS)
;EXPECTED DATA
;IS ROM PC CORRECT?
;BR IF YES
;ERROR, CRAM PC IS WRONG

; READ KMC11 REGISTERS
; LOAD POINTER
; CHECK FOR NO CHANGE
; SKIP LIST
```

```
4467 021210 BADHEAD
4468 :***** TEST 20 *****
4469 :*CRAM TEST OF NO JUMP(I) ON Z BIT CLEAR MICRO-PROCESSOR INSTRUCTION.
4470 :*CLEAR THE Z BIT, PERFORM THE JUMP INSTRUCTION.
4471 :*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
4472 :*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4473 :*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4474 :*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4475 :*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
4476 :*THEN PORT4 WILL CONTAIN A 37
4477 021210 BADHEAD
4478 :***** TEST 20 *****
4479
4480 021210 BGNTST
4481 021210 T20::
4482 021210
4483 021214 MYINT
4484 021220 004737 004222 MSTCLR :MASTER CLEAR M8200,4,6,7
4485 021224 JSR PC, MEMSET :SET MEM AND RAM
4486 021224 104404 1$: BGNSEG
4487 021226 TRAP CSBSEG
4488 021232 MSTCLR
4489 021236 100400 SR0MCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4490 021240 100400 :START AT ROM PC=0
4491 021244 115777 SR0MCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4492 021246 004737 004112 114377!<400*3> :JUMP TO ROM PC OF 1777
4493 021252 000001 JSR PC, RAMDAT :R4=CRAM PC (LSB 8 BITS)
4494 021254 020504 1 :EXPECTED DATA
4495 021256 001406 CMP R5,R4 :IS ROM PC CORRECT?
4496 021260 BEQ 2$ :BR IF YES
4497 021264 104455 ERROR 5 :ERROR, CRAM PC IS WRONG
4498 021266 000005 TRAP C$ERDF
4499 021270 005425 .WORD 5
4500 021272 007160 .WORD EMO
4501 021274 2$: ESCAPE SEG :.WORD ERR5
4502 021274 104410 TRAP C$ESCAPE
4503 021276 000002 .WORD 10000$-.
4504 021300 ENDSEG
4505 021300 10000$:
4506 021300 104405 TRAP C$ESEG
4507 021302 BGNSEG
4508 021302 104404 TRAP CSBSEG
4509 021304 000240 NOP
4510 021306 000240 NOP
4511 021310 004737 003750 JSR PC, CLRALL :CLEAR ALL CONDITIONS
4512 021314 SR0MCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4513 021320 100403 100403 :START AT ROM PC=3
4514 021322 SR0MCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4515 021326 101400 100000!<400*3> :JUMP TO ROM PC OF 0
4516 021330 004737 004112 JSR PC, RAMDAT :R4=CRAM PC (LSB 8 BITS)
4517 021334 000001 1 :EXPECTED DATA
4518 021336 020504 CMP R5,R4 :IS ROM PC CORRECT?
4519 021340 001406 BEQ 4$ :BR IF YES
4520 021342 ERROR 5 :ERROR, CRAM PC IS WRONG
4521 021346 104455 TRAP C$ERDF
4522 021350 000005 .WORD 5
```

```

4523 021352 005425      .WORD  EMO
4524 021354 007160      .WORD  ERR5
4525 021356          4$:  ESCAPE SEG
4526 021356 104410      TRAP   C$ESCAPE
4527 021360 000002      .WORD  10001$-.
4528 021362          ENDSEG
4529 021362          10001$:
4530 021362 104405      TRAP   C$ESEG
4531 021364          BGNSEG
4532 021364 104404      TRAP   C$BSEG
4533 021366 004737 003750  JSR    PC,CLRALL      ;CLEAR ALL CONDITIONS
4534 021372          SROMCLK  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4535 021376 100406      100406 ;START AT ROM PC=6
4536 021400          SROMCLK  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4537 021404 105525      104125!<400*3> ;JUMP TO ROM PC OF 525
4538 021406 004737 004112  JSR    PC,RAMDAT     ;R4=CRAM PC (LSB 8 BITS)
4539 021412 000001      1      ;EXPECTED DATA
4540 021414 020504      CMP    R5,R4        ;IS ROM PC CORRECT?
4541 021416 001406      BEQ    6$          ;BR IF YES
4542 021420          ERROR  5      ;ERROR, CRAM PC IS WRONG
4543 021424 104455      TRAP   C$ERDF
4544 021426 000005      .WORD  5
4545 021430 005425      .WORD  EMO
4546 021432 007160      .WORD  ERR5
4547 021434          6$:  ESCAPE SEG
4548 021434 104410      TRAP   C$ESCAPE
4549 021436 000002      .WORD  10002$-.
4550 021440          ENDSEG
4551 021440          10002$:
4552 021440 104405      TRAP   C$ESEG
4553 021442          CHKREG
4554          .LIST  ME
4555 021442 004737 003014  JSR    PC,RDKM      ; READ KMC11 REGISTERS
4556 021446 012702 021460  MOV    #64$,R2     ; LOAD POINTER
4557 021452 004737 003120  JSR    PC,RCHECK   ; CHECK FOR NO CHANGE
4558 021456 000405          BR    65$        ; SKIP LIST
4559 021460      000      000      000  64$: .BYTE .....20
4560 021463      000      000      000
4561 021466      000      000      000
4562 021471      020
4563 021472          65$:
4564 021472          ENDTST
4565 021472          L10065:
4566 021472 104401      TRAP   C$ETST

```



```

4567 021474      BADHEAD
4568             :***** TEST 21 *****
4569             :*CRAM TEST OF NO JUMP(I) ON BRO CLEAR MICRO-PROCESSOR INSTRUCTION.
4570             :*CLEAR THE BRO BIT, PERFORM THE JUMP INSTRUCTION.
4571             :*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
4572             :*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4573             :*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4574             :*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4575             :*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
4576             :*THEN PORT4 WILL CONTAIN A 37
4577 021474      BADHEAD
4578             :***** TEST 21 *****
4579
4580 021474      BGNTST
4581 021474      T21::
4582 021474
4583 021500      MYINT
4584 021504 004737 004222  MSTCLR           ;MASTER CLEAR M8200,4,6,7
4585 021510      JSR      PC, MEMSET          ;SET MEM AND RAM
4586 021510 104404 1$:  BGNSEG
4587 021512 004737 003750  TRAP      C$BSEG
4588 021516      JSR      PC, CLRALL         ;CLEAR ALL CONDITIONS
4589 021522 100400      SR0MCLK          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4590 021524      100400          ;START AT ROM PC=0
4591 021530 116377      SR0MCLK          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4592 021532 004737 004112  114377!<400*4> ;JUMP TO ROM PC OF 1777
4593 021536 000001      JSR      PC, RAMDAT        ;R4=CRAM PC (LSB 8 BITS)
4594 021540 020504      1          ;EXPECTED DATA
4595 021542 001406      CMP      R5, R4          ;IS ROM PC CORRECT?
4596 021544      BEQ      2$              ;BR IF YES
4597 021550 104455      ERROR          5          ;ERROR, CRAM PC IS WRONG
4598 021552 000005      TRAP      C$ERDF
4599 021554 005425      .WORD      5
4600 021556 007160      .WORD      EMO
4601 021560      .WORD      ERR5
4602 021560 104410 2$:  ESCAPE SEG
4603 021562 000002      TRAP      C$ESCAPE
4604 021564      .WORD      10000$-.
4605 021564      ENDSEG
4606 021564 104405 10000$: TRAP      C$ESEG
4607 021566      BGNSEG
4608 021566 104404      TRAP      C$BSEG
4609 021570 000240      NOP
4610 021572 000240      NOP
4611 021574 004737 003750  JSR      PC, CLRALL         ;CLEAR ALL CONDITIONS
4612 021600      SR0MCLK          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4613 021604 100403      100403          ;START AT ROM PC=3
4614 021606      SR0MCLK          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4615 021612 102000      100000!<400*4> ;JUMP TO ROM PC OF 0
4616 021614 004737 004112  JSR      PC, RAMDAT        ;R4=CRAM PC (LSB 8 BITS)
4617 021620 000001      1          ;EXPECTED DATA
4618 021622 020504      CMP      R5, R4          ;IS ROM PC CORRECT?
4619 021624 001406      BEQ      4$              ;BR IF YES
4620 021626      ERROR          5          ;ERROR, CRAM PC IS WRONG
4621 021632 104455      TRAP      C$ERDF
4622 021634 000005      .WORD      5

```

4623	021636	005425				.WORD	EMO		
4624	021640	007160				.WORD	ERR5		
4625	021642				4\$:	ESCAPE	SEG		
4626	021642	104410				TRAP	C\$ESCAPE		
4627	021644	000002				.WORD	10001\$-		
4628	021646					ENDSEG			
4629	021646				10001\$:				
4630	021646	104405				TRAP	C\$ESEG		
4631	021650					BGNSEG			
4632	021650	104404				TRAP	C\$BSEG		
4633	021652	004737	003750			JSR	PC,CLRALL		
4634	021656					SROMCLK			:CLEAR ALL CONDITIONS
4635	021662	100406				100406			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4636	021664					SROMCLK			:START AT ROM PC=6
4637	021670	106125				104125!<400*4>			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4638	021672	004737	004112			JSR	PC,RAMDAT		:JUMP TO ROM PC OF 525
4639	021676	000001				1			:R4=CRAM PC (LSB 2 BITS)
4640	021700	020504				CMP	R5,R4		:EXPECTED DATA
4641	021702	001406				BEQ	6\$:IS ROM PC CORRECT?
4642	021704					ERROR	5		:BR IF YES
4643	021710	104455				TRAP	C\$ERDF		:ERROR, CRAM PC IS WRONG
4644	021712	000005				.WORD	5		
4645	021714	005425				.WORD	EMO		
4646	021716	007160				.WORD	ERR5		
4647	021720				6\$:	ESCAPE	SEG		
4648	021720	104410				TRAP	C\$ESCAPE		
4649	021722	000002				.WORD	10002\$-		
4650	021724					ENDSEG			
4651	021724				10002\$:				
4652	021724	104405				TRAP	C\$ESEG		
4653	021726					CHKREG			
4654						.LIST	ME		
4655	021726	004737	003014			JSR	PC,RDKM		: READ KMC11 REGISTERS
4656	021732	012702	021744			MOV	#64\$,R2		: LOAD POINTER
4657	021736	004737	003120			JSR	PC,RCHECK		: CHECK FOR NO CHANGE
4658	021742	000405				BR	65\$: SKIP LIST
4659	021744	000	000	000	64\$:	.BYTE20		
4660	021747	000	000	000					
4661	021752	000	000	000					
4662	021755	020							
4663	021756				65\$:				
4664	021756				ENDTST				
4665	021756				L10066:				
4666	021756	104401				TRAP	C\$ETST		

```
4667 021760 BADHEAD
4668 :***** TEST 22 *****
4669 :*CRAM TEST OF NO JUMP(I) ON BR1 CLEAR MICRO-PROCESSOR INSTRUCTION.
4670 :*CLEAR THE BR1 BIT, PERFORM THE JUMP INSTRUCTION.
4671 :*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
4672 :*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4673 :*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4674 :*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4675 :*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
4676 :*THEN PORT4 WILL CONTAIN A 37
4677 021760 BADHEAD
4678 :***** TEST 22 *****
4679
4680 021760 BGNTST
4681 021760 T22::
4682 021760
4683 021764 MYINT
4684 021770 004737 004222 JSR PC, MEMSET ;MASTER CLEAR M8200,4,6,7
4685 021774 1$: BGNSEG ;SET MEM AND RAM
4686 021774 104404 TRAP C$BSEG
4687 021776 004737 003750 JSR PC, CLRALL ;CLEAR ALL CONDITIONS
4688 022002 SR0MCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4689 022006 100400 100400 ;START AT ROM PC=0
4690 022010 SR0MCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4691 022014 116777 114377!<400*5> ;JUMP TO ROM PC OF 1777
4692 022016 004737 004112 JSR PC, RAMDAT ;R4=CRAM PC (LSB 8 BITS)
4693 022022 000001 1 ;EXPECTED DATA
4694 022024 020504 CMP R5,R4 ;IS ROM PC CORRECT?
4695 022026 001406 BEQ 2$ ;BR IF YES
4696 022030 ERROR 5 ;ERROR, CRAM PC IS WRONG
4697 022034 104455 TRAP C$ERDF
4698 022036 000005 .WORD 5
4699 022040 005425 .WORD EMO
4700 022042 007160 .WORD ERR5
4701 022044 2$: ESCAPE SEG
4702 022044 104410 TRAP C$ESCAPE
4703 022046 000002 .WORD 10000$-.
4704 022050 ENDSEG
4705 022050 10000$:
4706 022050 104405 TRAP C$ESEG
4707 022052 BGNSEG
4708 022052 104404 TRAP C$BSEG
4709 022054 000240 NOP
4710 022056 000240 NOP
4711 022060 004737 003750 JSR PC, CLRALL ;CLEAR ALL CONDITIONS
4712 022064 SR0MCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4713 022070 100403 100403 ;START AT ROM PC=3
4714 022072 SR0MCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4715 022076 102400 100000!<400*5> ;JUMP TO ROM PC OF 0
4716 022100 004737 004112 JSR PC, RAMDAT ;R4=CRAM PC (LSB 8 BITS)
4717 022104 000001 1 ;EXPECTED DATA
4718 022106 020504 CMP R5,R4 ;IS ROM PC CORRECT?
4719 022110 001406 BEQ 4$ ;BR IF YES
4720 022112 ERROR 5 ;ERROR, CRAM PC IS WRONG
4721 022116 104455 TRAP C$ERDF
4722 022120 000005 .WORD 5
```



```

4723 022122 005425
4724 022124 007160
4725 022126
4726 022126 104410
4727 022130 000002
4728 022132
4729 022132
4730 022132 104405
4731 022134
4732 022134 104404
4733 022136 004737 003750
4734 022142
4735 022146 100406
4736 022150
4737 022154 106525
4738 022156 004737 004112
4739 022162 000001
4740 022164 020504
4741 022166 001406
4742 022170
4743 022174 104455
4744 022176 000005
4745 022200 005425
4746 022202 007160
4747 022204
4748 022204 104410
4749 022206 000002
4750 022210
4751 022210
4752 022210 104405
4753 022212
4754
4755 022212 004737 003014
4756 022216 012702 022230
4757 022222 004737 003120
4758 022226 000405
4759 022230 000 000 000 64$:
4760 022233 000 000 000
4761 022236 000 000 000
4762 022241 020
4763 022242
4764 022242
4765 022242
4766 022242 104401

        .WORD      EMO
        .WORD      ERR5
4$:     ESCAPE SEG
        TRAP      C$ESCAPE
        .WORD      10001$-.
        ENDSEG
10001$:
        TRAP      C$ESEG
        BGNSEG
        TRAP      C$BSEG
        JSR       PC,CLRALL
        SROMCLK
100406
        SROMCLK
104125! <400*5>
        JSR       PC,RAMDAT
        1
        CMP       R5,R4
        BEQ       6$
        ERROR     5
        TRAP      C$ERDF
        .WORD      5
        .WORD      EMO
        .WORD      ERR5
6$:     ESCAPE SEG
        TRAP      C$ESCAPE
        .WORD      10002$-.
        ENDSEG
10002$:
        TRAP      C$ESEG
        CHKREG
        .LIST     ME
        JSR       PC,RDKM
        MOV       #64$,R2
        JSR       PC,R$CHECK
        BR        65$
        .BYTE     .....20
; CLEAR ALL CONDITIONS
; NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
; START AT ROM PC=6
; NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
; JUMP TO ROM PC OF 525
; R4=CRAM PC (LSB 8 BITS)
; EXPECTED DATA
; IS ROM PC CORRECT?
; BR IF YES
; ERROR, CRAM PC IS WRONG

; READ KMC11 REGISTERS
; LOAD POINTER
; CHECK FOR NO CHANGE
; SKIP LIST

65$:
ENDTST
L10067:
        TRAP      C$ETST
    
```

```
4767 022244 BADHEAD
4768 :***** TEST 23 *****
4769 :*CRAM TEST OF NO JUMP(I) ON BR4 CLEAR MICRO-PROCESSOR INSTRUCTION.
4770 :*CLEAR THE BR4 BIT, PERFORM THE JUMP INSTRUCTION.
4771 :*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
4772 :*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4773 :*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4774 :*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4775 :*THE CRAM PC IS CORRECT, IF THE CRAM PC IS NOT RIGHT.
4776 :*THEN PORT4 CONTAINS A 37
4777 022244 BADHEAD
4778 :***** TEST 23 *****
4779
4780 022244 BGNTST
4781 022244 T23::
4782 022244
4783 022250
4784 022254 004737 004222 MYINT
4785 022260 104404 003750 JSR PC, MEMSET ;MASTER CLEAR M8200.4,6,7
4786 022260 104404 003750 1$: BGNSEG ;SET MEM AND RAM
4787 022262 004737 003750 TRAP C$BSEG
4788 022266 JSR PC, CLRALL ;CLEAR ALL CONDITIONS
4789 022272 100400 SROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4790 022274 100400 ;START AT ROM PC=0
4791 022300 117377 SROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4792 022302 004737 004112 114377! <400*6> ;JUMP TO ROM PC OF 1777
4793 022306 000001 JSR PC, RAMDAT ;R4=CRAM PC (LSB 8 BITS)
4794 022310 020504 1 ;EXPECTED DATA
4795 022312 001406 CMP R5, R4 ;IS ROM PC CORRECT?
4796 022314 BEQ 2$ ;BR IF YES
4797 022320 104455 ERROR 5 ;ERROR, CRAM PC IS WRONG
4798 022322 000005 TRAP C$ERDF
4799 022324 005425 .WORD 5
4800 022326 007160 .WORD EMO
4801 022330 2$: ESCAPE SEG ;.WORD ERR5
4802 022330 104410 TRAP C$ESCAPE
4803 022332 000002 .WORD 10000$-.
4804 022334 ENDSEG
4805 022334 10000$:
4806 022334 104405 TRAP C$ESEG
4807 022336 BGNSEG
4808 022336 104404 TRAP C$BSEG
4809 022340 000240 NOP
4810 022342 000240 NOP
4811 022344 004737 003750 JSR PC, CLRALL ;CLEAR ALL CONDITIONS
4812 022350 SROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4813 022354 100403 100403 ;START AT ROM PC=3
4814 022356 SROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4815 022362 103000 100000! <400*6> ;JUMP TO ROM PC OF 0
4816 022364 004737 004112 JSR PC, RAMDAT ;R4=CRAM PC (LSB 8 BITS)
4817 022370 000001 1 ;EXPECTED DATA
4818 022372 020504 CMP R5, R4 ;IS ROM PC CORRECT?
4819 022374 001406 BEQ 4$ ;BSR IF YES
4820 022376 ERROR 5 ;ERROR, CRAM PC IS WRONG
4821 022402 104455 TRAP C$ERDF
4822 022404 000005 .WORD 5
```

```

4823 022406 005425          .WORD  EMO
4824 022410 007160          .WORD  ERR5
4825 022412          4$:  ESCAPE  SEG
4826 022412 104410          TRAP   C$ESCAPE
4827 022414 000002          .WORD  10001$-.
4828 022416          ENDSEG
4829 022416          10001$:
4830 022416 104405          TRAP   C$ESEG
4831 022420          BGNSEG
4832 022420 104404          TRAP   C$BSEG
4833 022422 004737 003750          JSR    PC,CLRALL          ;CLEAR ALL CONDITIONS
4834 022426          SROMCLK          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4835 022432 100406          100406          ;START AT ROM PC=6
4836 022434          SROMCLK          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4837 022440 107125          104125!<400*6> ;JUMP TO ROM PC OF 525
4838 022442 004737 004112          JSR    PC,RAMDAT          ;R4=CRAM PC (LSB 8 BITS)
4839 022446 000001          1          ;EXPECTED DATA
4840 022450 020504          CMP    R5,R4          ;IS ROM PC CORRECT?
4841 022452 001406          BEQ    6$          ;BR IF YES
4842 022454          ERROR  5          ;ERROR, CRAM PC IS WRONG
4843 022460 104455          TRAP   C$ERDF
4844 022462 000005          .WORD  5
4845 022464 005425          .WORD  EMO
4846 022466 007160          .WORD  ERR5
4847 022470          6$:  ESCAPE  SEG
4848 022470 104410          TRAP   C$ESCAPE
4849 022472 000002          .WORD  10002$-.
4850 022474          ENDSEG
4851 022474          10002$:
4852 022474 104405          TRAP   C$ESEG
4853 022476          CHKREG
4854 022476          .LIST  ME
4855 022476 004737 003014          JSR    PC,RDKM          ; READ KMC11 REGISTERS
4856 022502 012702 022514          MOV    #64$,R2          ; LOAD POINTER
4857 022506 004737 003120          JSR    PC,RCHECK          ; CHECK FOR NO CHANGE
4858 022512 000405          BR     65$          ; SKIP LIST
4859 022514          000          000          000 64$: .BYTE .....20
4860 022517          000          000
4861 022522          000          000
4862 022525          020
4863 022526          65$:
4864 022526          ENDTST
4865 022526          L10070:
4866 022526 104401          TRAP   C$SETST

```



```

4867 022530          BADHEAD
4868                :***** TEST 24 *****
4869                :*CRAM TEST OF NO JUMP(I) ON BR7 CLEAR MICRO-PROCESSOR INSTRUCTION.
4870                :*CLEAR THE BR7 BIT, PERFORM THE JUMP INSTRUCTION.
4871                :*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
4872                :*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4873                :*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4874                :*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4875                :*THE CRAM PC IS CORRECT, IF THE CRAM PC IS NOT RIGHT.
4876                :*THEN PORT4 CONTAINS A 37
4877 022530          BADHEAD
4878                :***** TEST 24 *****
4879
4880 022530          BGNTST
4881 022530          T24::
4882 022530
4883 022534          MYINT
4884 022540 004737 004222  MSTCLR                :MASTER CLEAR M8200,4,6,7
4885 022544          JSR      PC, MEMSET                :SET MEM AND RAM
4886 022544 104404          1$:  BGNSEG
4887 022546 004737 003750  TRAP      C$BSEG
4888 022552          JSR      PC, CLRALL                :CLEAR ALL CONDITIONS
4889 022556 100400          SR0MCLK                :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4890 022560          100400          :START AT ROM PC=0
4891 022564 117777          SR0MCLK                :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4892 022566 004737 004112  114377! <400*7>  JSR      PC, RAMDAT                :JUMP TO ROM PC OF 1777
4893 022572 000001          1          :R4=CRAM PC (LSB 8 BITS)
4894 022574 020504          CMP      R5,R4                :EXPECTED DATA
4895 022576 001406          BEQ      2$                :IS ROM PC CORRECT?
4896 022600          ERROR 5                :BR IF YES
4897 022604 104455          TRAP      C$ERDF                :ERROR, CRAM PC IS WRONG
4898 022606 000005          .WORD 5
4899 022610 005425          .WORD  EMO
4900 022612 007160          .WORD  ERR5
4901 022614          2$:  ESCAPE SEG
4902 022614 104410          TRAP      C$ESCAPE
4903 022616 000002          .WORD  10000$-.
4904 022620          ENDSEG
4905 022620          10000$:
4906 022620 104405          TRAP      C$ESEG
4907 022622          BGNSEG
4908 022622 104404          TRAP      C$BSEG
4909 022624 000240          NOP
4910 022626 000240          NOP
4911 022630 004737 003750  JSR      PC, CLRALL                :CLEAR ALL CONDITIONS
4912 022634          SR0MCLK                :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4913 022640 100403          100403          :START AT ROM PC=3
4914 022642          SR0MCLK                :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4915 022646 103400          100000! <400*7>  :JUMP TO ROM PC OF 0
4916 022650 004737 004112  JSR      PC, RAMDAT                :R4=CRAM PC (LCB 8 BITS)
4917 022654 000001          1          :EXPECTED DATA
4918 022656 020504          CMP      R5,R4                :IS ROM PC CORRECT?
4919 022660 001406          BEQ      4$                :BR IF YES
4920 022662          ERROR 5                :ERROR, CRAM PC IS WRONG
4921 022666 104455          TRAP      C$ERDF
4922 022670 000005          .WORD  5

```

4923	022672	005425					.WORD	EMO			
4924	022674	007160					.WORD	ERR5			
4925	022676					4\$:	ESCAPE	SEG			
4926	022676	104410					TRAP	C\$ESCAPE			
4927	022700	000002					.WORD	10001\$-			
4928	022702						ENDSEG				
4929	022702					10001\$:					
4930	022702	104405					TRAP	C\$ESEG			
4931	022704						BGNSEG				
4932	022704	104404					TRAP	C%BSEG			
4933	022706	004737	003750				JSR	PC,CLRALL			:CLEAR ALL CONDITIONS
4934	022712						SROMCLK				:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4935	022716	100406					100406				:START AT ROM PC=6
4936	022720						SROMCLK				:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4937	022724	107525					104125!<400*7>				:JUMP TO ROM PC OF 525
4938	022726	004737	004112				JSR	PC,RAMDAT			:R4=CRAM PC (LSB 8 BITS)
4939	022732	000001					1				:EXPECTED DATA
4940	022734	020504					CMP	R5,R4			:IS ROM PC CORRECT?
4941	022736	001406					BEQ	6\$:BR IF YES
4942	022740						ERROR	5			:ERROR, CRAM PC IS WRONG
4943	022744	104455					TRAP	C\$ERDF			
4944	022746	000005					.WORD	5			
4945	022750	005425					.WORD	EMO			
4946	022752	007160					.WORD	ERR5			
4947	022754					6\$:	ESCAPE	SEG			
4948	022754	104410					TRAP	C\$ESCAPE			
4949	022756	000002					.WORD	10002\$-			
4950	022760						ENDSEG				
4951	022760					10002\$:					
4952	022760	104405					TRAP	C\$ESEG			
4953	022762						CHKREG				
4954							.LIST	ME			
4955	022762	004737	003014				JSR	PC,RDKM			: READ KMC11 REGISTERS
4956	022766	012702	023000				MOV	#64\$,R2			: LOAD POINTER
4957	022772	004737	003120				JSR	PC,RCHECK			: CHECK FOR NO CHANGE
4958	022776	000405					BR	65\$: SKIP LIST
4959	023000	000	000	000	000	64\$:	.BYTE,20			
4960	023003	000	000	000	000						
4961	023006	000	000	000	000						
4962	023011	020									
4963	023012					65\$:					
4964	023012					ENDTST					
4965	023012					L10071:					
4966	023012	104401					TRAP	C\$ETST			

4967 023014
4968
4969
4970
4971
4972
4973
4974
4975 023014
4976
4977
4978 023014
4979 023014
4980 023014
4981 023020
4982 023024 005002
4983 023026 042737 000037 023052 1\$:
4984 023034 050237 023052
4985 023040
4986 023044 010000
4987 023046
4988 023052 004000 2\$:
4989
4990 023054 010261 000004
4991 023060
4992 023064 122500
4993 023066 005202
4994 023070 032702 000020
4995 023074 001754
4996
4997
4998
4999
5000
5001
5002 023076 005002
5003
5004 023100 042737 000037 023116 3\$:
5005 023106 050237 023116
5006 023112
5007 023116 004000 4\$:
5008 023120
5009 023124 041224
5010 023126 116104 000004
5011 023132 110205
5012 023134 120504
5013 023136 001406
5014
5015 023140
5016 023144 104455
5017 023146 000015
5018 023150 005425
5019 023152 010364
5020
5021
5022

BADHEAD

:***** TEST 25 *****

:*

:*MAIN MEMORY PAGE DUAL ADDRESS TEST.
:*IN THIS TEST WE WILL VERIFY THAT PAGES DO
:*NOT DUAL ADDRESS. THIS TEST IS DIFFERENT FROM THE
:*PREVIOUS DUAL ADDRESS TESTS IN THAT THE OTHER
:*TEST REALLY DIDN'T CHECK PAGE DUAL ADDRESSING

BADHEAD

:***** TEST 25 *****

BGNTST

T25::

MYINT

MSTCLR

CLR R2

BIC #37,2\$

BIS R2,2\$

ROMCLK

10000

ROMCLK

4000

MOV R2,4(R1)

ROMCLK

122500

INC R2

BIT #20,R2

BEQ 1\$

:R2 WILL BE PAGE #
:CLEAR UNUSED BITS
:ADD CURRENT PAGE MARKER.
:SET ADDR D

:OF PAGE X
:THIS LOCATION MODIFIED BY LOST
:FEW INSTRUCTIONS
:PUT PAGE # INTO PART 4
:CLOCK PART 4 INTO MEMORY
:WHOSE PAGE # IS IN R2
:UPDATE PAGE #
:DONE ALL PAGES?
:NO-DO NEXT ONE

:OK, ALL LOADS OF ALL PAGES
:CONTAIN THIER PAGE NUMBER, NOW
:WE'LL GO BACK THROUGH
:THEM CHECKING THEM OUT.

:R2 STILL HAS PAGE NUMBER

CLR R2

BIC #37,4\$

BIS R2,4\$

ROMCLK

4000

ROMCLK

041224

MOVB 4(R1),R4

MOVB R2,R5

CMPB R5,R4

BEQ 5\$

:LOAD PAGE NUMBER

:MOVE MEM TO PART 4

: 'FOUND'
: 'EXPECTED'
: ADDRESS PROBLEM?

:PAGE ADDRESSING ERROR IN MAIN

ERROR 13

TRAP CSERDF

.WORD 13

.WORD EMO

.WORD ERR13

:MEMORY.
:MAR BITS 8 THROUGH 12 ARE REPRESENTED
:BY R2 ('EXP'ED')BITS 0-4)


```
5023
5024 023154          5$:  ESCAPE TST
5025 023154 104410   TRAP  C$ESCAPE
5026 023156 000042   .WORD L10072-.
5027 023160 005202   INC R2           :UPDATE PAGE ADDRESS
5028 023162 032702 000020 BIT #20,R2      :ALL DONE?
5029 023166 001744   BEQ 3$          :NO-CHECK NEXT PAGE.
5030                                         :YES-EXIT.
5031 023170          CHKREG
5032          .LIST  ME
5033 023170 004737 003014 JSR  PC,RDKM     : READ KMC11 REGISTERS
5034 023174 012702 023206 MOV  #64$,R2     : LOAD POINTER
5035 023200 004737 003120 JSR  PC,RCHECK   : CHECK FOR NO CHANGE
5036 023204 000405          BR    65$       : SKIP LIST
5037 023206          000    000    000 64$: .BYTE .....20
5038 023211          000    000    000
5039 023214          000    000    000
5040 023217          020
5041 023220          65$:
5042 023220          ENDTST
5043 023220          L10072:
5044 023220 104401   TRAP  C$ETST
```

```
5045 023222 BADHEAD
5046 :***** TEST 26 *****
5047 :*
5048 :*JUMP FIELD,PAGE TEST
5049 :*
5050 :*IN THIS TEST WILL MAKE SURE A JUMP FIELD INSTRUCTION
5051 :*WORKS. TO DO THIS, WE'LL PUT THE DESIRED PAGE,FIELD
5052 :*INFORMATION IN IBUS*<13> THEN ISSUE A JUMP FIELD
5053 :*THEN WE'LL READ PC REG. AND VERIFY.
5054 023222 BADHEAD
5055 :***** TEST 26 *****
5056
5057 023222 BGNTST
5058 023222 T26::
5059 023222 MYINT
5060 023222 MSTCLR
5061
5062 023232 005002 CLR R2 ;R2 TO CONTAIN FIELD #
5063
5064 023234 042737 000017 023252 1$: BIC #17,2$ ;CLEAR ANY JUNK
5065 023242 050237 023252 BIS R2,2$ ;SET FIELD # INTO INSTR.
5066
5067 023246 ROMCLK ;CLOCK FIELD BITS INTO BREG.
5068 023252 000400 2$: 000400 ;CONTAINS FIELD,PAGE BITS
5069 023254 ROMCLK ;XFERR BREG INTO IBUS*13>
5070 023260 061233 061233
5071 023262 SRMCLK ;GET INSTRUCTION CLOCKED.
5072 023266 100000 100000 ;BAS FORM FOR JUM FIELD INSTR.
5073
5074
5075 023270 142761 000002 000001 BICB #BIT1,1(R1) ;CLEAR ROMI
5076 023276 ROMCLK ;CLOCK NEXT INSTR.
5077 023302 121264 121264 ;MOVE IBUS*TO PORT 4
5078 023304 116104 000004 ;GET IT.
5079 023310 042704 177760 BIC #^C<17>,R4
5080 023314 120402 CMPB R4,R2 ;FIELD OK?
5081 023316 001407 BEQ 3$ ;IF OK GO AHEAD
5082 023320 010205 MOV R2,R5
5083 023322 ERROR 12 ;CHANGE FIELD INSTRUCTION
5084 023326 104455 TRAP C$ERDF
5085 023330 000014 .WORD 12
5086 023332 005425 .WORD EMO
5087 023334 010212 .WORD ERR12
5088
5089 ;FAILED. FOR FIELD,PAGE INDICATES
5090 ;BY "EXPECTED" BITS 0,1,2,3 OF
5091 ;EXPECTED REPRESENT FIELD BITS.
5091 023336 3$: ESCAPE TST
5092 023336 104410 TRAP C$ESCAPE
5093 023340 000042 .WORD L10073-.
5094
5095
5096 023342 005202 INC R2 ;UPDATE TO NEXT FIELD
5097 023344 032702 000020 BIT #20,R2 ;DONE ALL FIELDS?
5098 023350 001731 BEQ 1$
5099
5100 023352 CHKREG
```

5101							.LIST	ME		
5102	023352	004737	003014				JSR	PC,RDKM	:	READ KMC11 REGISTERS
5103	023356	012702	023370				MOV	#64\$,R2	:	LOAD POINTER
5104	023362	004737	003120				JSR	PC,RCHECK	:	CHECK FOR NO CHANGE
5105	023366	000405					BR	65\$:	SKIP LIST
5106	023370	000	000	000	64\$:		.BYTE20		
5107	023373	000	000	000						
5108	023376	000	000	000						
5109	023401	020								
5110	023402				65\$:					
5111	023402				ENDTST					
5112	023402				L10073:					
5113	023402	104401			TRAP	C\$ETST				

5114 023404
5115
5116
5117
5118
5119
5120
5121
5122
5123
5124
5125
5126
5127
5128
5129
5130
5131
5132
5133
5134
5135
5136
5137
5138
5139
5140
5141
5142
5143
5144
5145
5146
5147 023404
5148
5149
5150 023404
5151 023404
5152 023404
5153 023410
5154
5155 023414 012737 000000 023530
5156 023422 012737 000000 002354
5157
5158 023430
5159 023430 113737 002355 023450
5160 023436 113737 002354 023464
5161
5162
5163 023444
5164 023444
5165 023450 000400
5166 023452
5167 023456 061233
5168 023460
5169 023464 100000

```

BADHEAD
;***** TEST 27 *****
:*
:*JUMP TEST, JUMP ALWAYS, JUMP CHANGE FIELD
:*
:*IN THIS TEST, WE WILL CHECK THE ABILITY OF THE
:*MICRO PROCESSOR TO JUMP (BRANCH & ALWAYS INSTRUCTION)
:*TO LOCATIONS, FIELDS FROM OTHER LOCATIONS FIELDS.
:*WE ALREADY KNOW THAT THE BRANCH INSTR WORKS FROM
:*OTHER TEST.
PROCEDURE:
1. START ADDR 0, FIELD 0
2. **CALCULATE NEW ADDR, FIELD VIA INC,
3. CAUSE JUMP (BRANCH) TO NEW ADDRESS
4. READ PC FROM IBUS*12 AND IBUS*13
5. REPEAT STEP 2-4 256.TIMES

TO CALCULATE NEW ADDRESS:
1. INC LOW BYTE OF ADDRESS FOR PC ADDRESS 0-7
2. INC LOW BYTE OF NADDRESS FOR PC ADDRESS 8-11
   BITS REPRESENTED AS BITS 0-3. WHEN 0-3 OVERFLOWS,
   RESTARTS AT ZERO.

NET RESULT IS JUMPS FROM:
FIELD,PAGE          LOC
0                    0
1                    1
2                    3
3                    7
10                   11
11                   :
:TO                  :
17                   377

BADHEAD
;***** TEST 27 *****

T27::
BGNTST
MYINT
MSTCLR

1$:
MOV #0,55$          ; 55$= DESTINATION ADDRESS
MOV #0,FADR         ; SOURCE ADDRESS
;LOOP HERE

MOV B FADR+1,2$    ; LOAD IMM INST. FOR PC-HIGH
MOV B FADR,3$      ; DITTO FOR PC-LOW

11$:
ROMCLK
000400
2$:
ROMCLK             ;MOVE PAGE,FIELD # TO BREG.
61233
;MOV BREG TO PC HIGH REG.
SROMCLK
3$:
100000             ;NOW CLOCK IT IN BY JMP FIELD INSTR.
    
```

```

5170 023466 113737 023531 023506      MOVB 55$+1,22$      : FIELD, PAGE FOR DEST.
5171 023474 113737 023530 023522      MOVB 55$,33$      : PC 7-0
5172 023502                                     ROMCLK
5173 023506 000400      22$: 000400
5174 023510                                     ROMCLK
5175 023514 061233      61233
5176 023516                                     SRCLK
5177 023522 100000      33$: 100000      : BR TO DEST.
5178 023524 004737 004112      JSR PC,RAMDAT     : GET PC
5179 023530 000000      55$: 0            : DEST. ADDRESS GOES HERE
5180 023532      5$:
5181 023532 020504      CMP R5,R4        : JUMP GO OK?
5182 023534 001406      BEQ 4$          : YEA, CONTINUES
5183 023536      ERROR 15      : FAILED TO JUMP PROPERLY.
5184 023542 104455      TRAP C$ERDF
5185 023544 000017      .WORD 15
5186 023546 005425      .WORD EMO
5187 023550 010710      .WORD ERR15

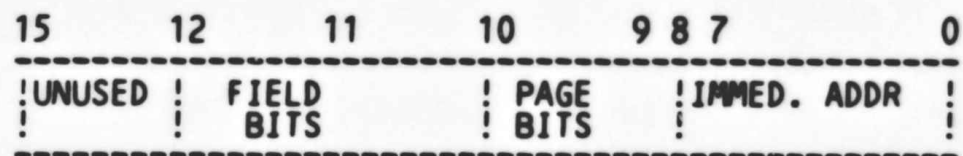
```

```

: "FROM ADDR" REPRESENTS
: THE ADDRESS WE STARTED AT
: "TO ADDR" REPRESENTS WHERE
: WE EXPECTED TO JUMP TO,
: "BAD ADDR" REPRESENTS WHERE
: WE WENT TO.

```

.REM %



```

;THIS IS A PICTURE OF THE P.C. REG.
BITS 0-7 ARE IN IBUS*<12>
BITS 8-11 ARE IN IBUS*<13>
THEY GOT CLOCK IN THERE VIA JUMPS TAKEN
THE FIELD BITS
ARE IN BIT POSITION 0,1 OF THE INSTRUCTION AT 2$.

```

```

3$ WAS THE JUMP ALWAYS INSTRUCTION. THE IMMED. ADDR.
WAS IN 0-7 OF THE JUMP INSTR. THE PAGE BITS,
PC REG BITS 8,9, WERE IN BITS 11,12 OF THE INSTR.
JUMP INSTRUCTIONS HAVE BEEN CHECKED OUT
BEFORE, SO THE IMPORTANT THING TO REMEMBER TO
WATCH IS THE "FROM ADDR", "TO ADDR"

```

%

```

5218 023552                                     ESCAPE TST
5219 023552 104410      TRAP C$ESCAPE
5220 023554 000166      .WORD L10074-.
5221 023556 005237 023530      INC 55$          : BUMP DEST.
5222 023562 032737 010000 023530      BIT #BIT12,55$  : DONE 4K?
5223 023570 001725      BEQ 11$         : DO NEXT IF NO
5224 023572 005037 023530      CLR 55$        : YES, RESET AND GO TO NEXT PAGE
5225 023576 005237 002354      INC FADR       : BUMP PAGE

```

```

5226 023602 113737 002354 002355      MOVB   FADR,FADR+1      : MAKE 12 BIT ADDRESS
5227 023610 032737 010000 002354      BIT    #BIT12,FADR     : DONE?
5228 023616 001704                      BEQ    1$              : DO AGAIN IF NO
5229
5230
5231
5232
5233
5234 023620 000240                      NOP
5235 023622 000240                      NOP
5236 023624 005011                      CLR    (R1)            : CLEAR RUN, IF UP
5237 023626 052711 040000      BIS    #40000,(R1)     : SET MASTER CLEAR
5238 023632                      SROMCLK                : WE MUST FIRST CLOCK
5239 023636 121265      121265      SROMCLK                : THE PC LATCH REGS
5240 023640                      121265      SROMCLK                : BEFORE WE CAN READ THEM
5241 023644 121244      121244      ROMCLK
5242 023646                      121244      ROMCLK                : REG PC REG HI, PUT IN PORT5
5243 023652 121265      121265      ROMCLK
5244 023654                      121265      ROMCLK                : REG PC REG LOW, PUT IN PORT4
5245 023660 121244      121244      CLR    R5              : EXPECT ZERO
5246 023662 005005                      MOV    4(R1),R4        : READ PC REG FROM PORT 4&5
5247 023664 016104 000004      BIC    #170001,R4
5248 023670 042704 170001      BEQ    40$
5249 023674 001406
5250
5251
5252
5253
5254
5255
5256 023676                      ERROR  45              : IF CLEARED, EXIT
5257 023702 104455      TRAP   C$ERDF          : NOTE WE ALSO CLEARED BIT 1 OF THE
5258 023704 000055      .WORD  45              : PC REG, BECAUSE AFTER THE MASTER
5259 023706 005425      .WORD  EMO             : CLEAR, WE DID TWO INSTRUCTIONS TO
5260 023710 012246      .WORD  ERR45           : READ IT, THUS CAUSING THE PC REG
5261
5262 023712                      :PC REG
5263 023712
5264
5265 023712 004737 003014      CHKREG
5266 023716 012702 023730      .LIST  ME
5267 023722 004737 003120      JSR    PC,RDKM         : READ KMC11 REGISTERS
5268 023726 000405                      MOV    #64$,R2        : LOAD POINTER
5269 023730 000 000 000 64$:  BR     65$            : CHECK FOR NO CHANGE
5270 023733 000 000 000      .BYTE  .....20      : SKIP LIST
5271 023736 000 000 000
5272 023741 020
5273 023742
5274 023742
5275 023742
5276 023742 104401      65$:
                    ENDTST
                    L10074:
                    TRAP   C$ETST

```



```

5277 023744 BADHEAD
5278 :***** TEST 28 *****
5279 :*
5280 :* IN THIS TEST WE'LL VERIFY THAT THE Z BIT CAN BE READ FROM
5281 :* IBUS*<13>. WE ALLREADY KNOW THAT THE Z BIT WORKS PROPERLY,
5282 :* ALL WE WANT TO KNOW HERE IS THAT IT CAN BE READ.
5283 :*
5284 023744 BADHEAD
5285 :***** TEST 28 *****
5286
5287 023744 BGNTST
5288 023744 T28:: MSTCLR
5289 023744 MYINT
5290 023750 JSR PC,CLRALL ;CLR CONDITION CODES.
5291 023754 004737 003750 ROMCLK ;NOW READ IBUS*<15>PUT IN PORT 4
5292 023760 121264 121264 MOVB 4(R1),R4 ;READ IT FROM PORT 4
5293 023764 121264 121264 BIC #177477,R4 ;STRIP ANY JUNK,C&Z BITS 6,7
5294 023766 116104 000004 MOV #0,R5 ;EXPECT IT CLEAR
5295 023772 042704 177477 CMPB R4,R5 ;OK?
5296 023776 012705 000000 BEQ 1$ ;FAILURE OF Z&C TO BE CLEAR.
5297 024002 120405 ERROR 16
5298 024004 001410 TRAP C$ERDF
5299 024006 .WORD 16
5300 024012 104455 .WORD EMO
5301 024014 000020 .WORD ERR16
5302 024016 005425
5303 024020 011066
5304
5305 024022 ESCAPE TST
5306 024022 104410 TRAP C$ESCAPE
5307 024024 000146 .WORD L10075-
5308 024026 004737 004074 1$: JSR PC,SETZ ;SET Z BIT.
5309 024032 ROMCLK ;NOW GO BACK AND CHECK Z BIT SET.
5310 024036 121264 121264
5311
5312 024040 016104 000004 MOV 4(R1),R4 ;GET INFO.
5313 024044 042704 177477 BIC #^C<300>,R4 ;STRIP FOR C&Z BITS.
5314 024050 012705 000200 MOV #200,R5 ;EXPECT ONLY Z BIT SET.
5315 024054 120405 CMPB R4,R5 ;SET OK?
5316 024056 001410 BEQ 2$ ;Z BIT FAILED TO SET PROPERLY.
5317 024060 ERROR 16
5318 024064 104455 TRAP C$ERDF
5319 024066 000020 .WORD 16
5320 024070 005425 .WORD EMO
5321 024072 011066 .WORD ERR16
5322
5323 024074 ESCAPE TST
5324 024074 104410 TRAP C$ESCAPE
5325 024076 000074 .WORD L10075-
5326 024100 004737 003750 2$: JSR PC,CLRALL ;NOW TRY TO CLEAR Z BIT.
5327 024104 ROMCLK
5328 024110 121264 121264
5329 024112 016104 000004 MOV 4(R1),R4
5330 024116 042704 177477 BIC #^C<300>,R4 ;STRIP FOR C&Z BITS
5331 024122 001407 BEQ 3$ ;IF ZERO,WE'RE OK
5332 024124 005005 CLR R5 ;ELSE REPORT ERROR

```

```
5333 024126          ERROR 16          ;Z BIT FAILED TO CLEAR PROPERLY.
5334 024132 104455   TRAP  CSERDF
5335 024134 000020   .WORD 16
5336 024136 005425   .WORD EMO
5337 024140 011066   .WORD ERR16
5338 024142          3$:
5339 024142          CHKREG
5340          .LIST ME
5341 024142 004737 003014   JSR PC,RDKM      : READ KMC11 REGISTERS
5342 024146 012702 024160   MOV #64$,R2     : LOAD POINTER
5343 024152 004737 003120   JSR PC,RCHECK  : CHECK FOR NO CHANGE
5344 024156 000405          BR 65$         : SKIP LIST
5345 024160          000 000 000 64$: .BYTE .....20
5346 024163          000 000 000
5347 024166          000 000 000
5348 024171          020
5349 024172          65$:
5350 024172          ENDTST
5351 024172          L10075:
5352 024172 104401   TRAP  C$ETST
```

```

5353 024174          BADHEAD
5354                :***** TEST 29 *****
5355                :*
5356                :* IN THIS TEST WE'LL VERIFY THAT THE C BIT CAN BE READ FROM
5357                :* IBUS*<13>. WE ALLREADY KNOW THAT THE C BIT WORKS PROPERLY,
5358                :* ALL WE WANT TO KNOW HERE IS THAT IT CAN BE READ.
5359                :*
5360 024174          BADHEAD
5361                :***** TEST 29 *****
5362
5363 024174          BGNTST
5364 024174          T29::
5365 024174          K4ONLY                ;M8206 &M8207 ONLY!
5366 024204 104432  TRAP          C$EXIT
5367 024206 000230  .WORD        L10076-.
5368 024210          MSTCLR
5369 024214          MYINT
5370 024220 004737 003750  JSR          PC,CLRALL          ;CLR CONDITION CODES.
5371 024224          ROMCLK                ;NOW READ IBUS*<13>PUT IN PORT 4
5372 024230 121264 121264
5373 024232 116104 000004  MOVB         4(R1),R4          ;READ IT FROM PORT 4
5374 024236 042704 177477  BIC          #177477,R4      ;STRIP ANY JUNK,C&Z BITS 6,7
5375 024242 012705 000000  MOV          #0,R5           ;EXPECT IT CLEAR
5376 024246 120405          CMPB         R4,R5           ;OK?
5377 024250 001410          BEQ          1$
5378 024252          ERROR          16          ;FAILURE OF Z&C TO BE CLEAR.
5379 024256 104455          TRAP          C$ERDF
5380 024260 000020          .WORD        16
5381 024262 005425          .WORD        EMO
5382 024264 011066          .WORD        ERR16
5383
5384 024266          ESCAPE TST
5385 024266 104410          TRAP          C$ESCAPE
5386 024270 000146          .WORD        L10076-.
5387 024272 004737 004042  JSR          PC,SETC          ;SET C BIT.
5388 024276          ROMCLK                ;NOW GO BACK AND CHECK C BIT SET.
5389 024302 121264 121264
5390 024304 016104 000004  MOV          4(R1),R4          ;GET INFO.
5391 024310 042704 177477  BIC          #^C<300>,R4     ;STRIP FOR C&Z BITS.
5392 024314 012705 000100  MOV          #100,R5         ;EXPECT ONLY C BIT SET.
5393 024320 120405          CMPB         R4,R5           ;SET OK?
5394 024322 001410          BEQ          2$
5395 024324          ERROR          16          ;C BIT FAILED TO SET PROPERLY.
5396 024330 104455          TRAP          C$ERDF
5397 024332 000020          .WORD        16
5398 024334 005425          .WORD        EMO
5399 024336 011066          .WORD        ERR16
5400
5401 024340          ESCAPE TST
5402 024340 104410          TRAP          C$ESCAPE
5403 024342 000074          .WORD        L10076-.
5404 024344 004737 003750  JSR          PC,CLRALL          ;NOW TRY TO CLEAR C BIT.
5405 024350          ROMCLK                ;NOW TRY TO CLEAR C BIT.
5406 024354 121264 121264
5407 024356 016104 000004  MOV          4(R1),R4          ;STRIP FOR C&Z BITS
5408 024362 042704 177477  BIC          #^C<300>,R4

```


5409	024366	001407					BEQ	3\$:IF ZERO,WE'RE OK
5410	024370	005005					CLR	R5		:ELSE REPORT ERROR
5411	024372						ERROR	16		:C BIT FAILED TO CLEAR PROPERLY.
5412	024376	104455					TRAP	C\$ERDF		
5413	024400	000020					.WORD	16		
5414	024402	005425					.WORD	EM0		
5415	024404	011066					.WORD	ERR16		
5416	024406					3\$:				
5417	024406						CHKREG			
5418							.LIST	ME		
5419	024406	004737	003014				JSR	PC,RDKM		: READ KMC11 REGISTERS
5420	024412	012702	024424				MOV	#65\$,R2		: LOAD POINTER
5421	024416	004737	003120				JSR	PC,RCHECK		: CHECK FOR NO CHANGE
5422	024422	000405					BR	66\$: SKIP LIST
5423	024424	000	000	000	65\$:		.BYTE20		
5424	024427	000	000	000						
5425	024432	000	000	000						
5426	024435	020								
5427	024436					66\$:				
5428	024436						ENDTST			
5429	024436				L10076:					
5430	024436	104401					TRAP	C\$ETST		

```
5431 024440 BADHEAD
5432 :***** TEST 30 *****
5433 :*TEST OF PROGRAM CLOCK BIT
5434 :*DO A MASTER CLEAR, VERIFY THAT PROGRAM CLOCK IS SET
5435 :*WRITE PROGRAM CLOCK BIT TO A ONE, VERIFY THAT IT CLEARS,
5436 :*AND THEN SETS SOME TIME LATER
5437 024440 BADHEAD
5438 :***** TEST 30 *****
5439
5440 024440 BGNTST
5441 024440 T30::
5442 024440
5443 024444 MYINT
5444 024450 005037 002402 MSTCLR ;MASTER CLEAR M8200,4,6,7
5445 024454 005037 002406 CLR TEMP ;PREPARE FOR
5446 024460 012702 000011 CLR $TMP0 ;DELAY
5447 024464 MOV #11,R2 ;SAVE FOR TYPEOUT
5448 024470 121224 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5449 024472 016104 000004 121224 ;PORT 4 LU11
5450 024476 042704 000357 MOV 4(R1),R4 ;PUT 'FOUND' IN R4
5451 024502 012737 000020 002414 BIC #357,R4 ;CLEAR UNWANTED BITS
5452 024510 123704 002414 MOV #20,$GDDAT ;PUT 'EXPECTED' IN $GDDAT
5453 024514 001406 CMPB $GDDAT,R4 ;IS PGM CLOCK SET?
5454 024516 BEQ 1$ ;
5455 024522 104455 ERROR 14 ;ERROR, PGM CLOCK IS NOT SET
5456 024524 000016 TRAP C$ERDF
5457 024526 005425 .WORD 14
5458 024530 010536 .WORD EMO
5459 024532 012761 000020 000004 1$: MOV #20,4(R1) ;LOAD PORT 4
5460 024540 152761 000002 000001 BISB #BIT1,1(R1) ;SET ROMI
5461 024546 012761 121111 000006 MOV #121111,6(R1) ;SEL6 INSTRUCTION
5462 024554 152761 000003 000001 BISB #BIT1!BIT0,1(R1) ;SET CLOCK BIT
5463 024562 012761 121224 000006 MOV #121224,6(R1) ;LOAD NEXT INSTRUCTION
5464 024570 152761 000003 000001 BISB #BIT1!BIT0,1(R1) ;READ CLOCK BIT
5465 024576 142761 030001 000001 BICB #BIT!BIT0,1(R1) ;CLEAR MAINT BITS
5466 024604 016104 000004 MOV 4(R1),R4 ;PUT 'FOUND' IN R4
5467 024610 005037 002414 CLR $GDDAT ;PUT 'EXPECTED' IN $GDDAT
5468 024614 123704 002414 CMPB $GDDAT,R4 ;IS PGM CLOCK CLEAR?
5469 024620 001406 BEQ 2$ ;
5470 024622 ERROR 14 ;ERROR, PGM CLOCK IS NOT CLEAR
5471 024626 104455 TRAP C$ERDF
5472 024630 000016 .WORD 14
5473 024632 005425 .WORD EMO
5474 024634 010536 .WORD ERR14
5475 024636 2$:
5476 024636 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5477 024642 121224 121224 ;PORT4 LU11
5478 024644 122761 000020 000004 CMPB #20,4(R1) ;IS PGM CLOCK SET?
5479 024652 001416 BEQ 3$ ;BR IF YES
5480 024654 005237 002402 INC TEMP ;INCREMENT DELAY
5481 024660 005537 002406 ADC $TMP0 ;INCREMENT DELAY
5482 024664 022737 000006 002406 CMP #6,$TMP0 ;IS DELAY DONE
5483 024672 001361 BNE 2$ ;BR IF NO
5484 024674 ERROR 14 ;ERROR PGM CLOCK NOT SET
5485 024700 104455 TRAP C$ERDF
5486 024702 000016 .WORD 14
```

5487	024704	005425					.WORD	EMO	
5488	024706	010536					.WORD	ERR14	
5489	024710					3\$:			
5490									
5491	024710						CHKREG		
5492							.LIST	ME	
5493	024710	004737	003014				JSR	PC,RDKM	: READ KMC11 REGISTERS
5494	024714	012702	024726				MOV	#64\$,R2	: LOAD POINTER
5495	024720	004737	003120				JSR	PC,RCHECK	: CHECK FOR NO CHANGE
5496	024724	000405					BR	65\$: SKIP LIST
5497	024726	000	000	000	64\$:		.BYTE,20	
5498	024731	000	000	000					
5499	024734	000	000	000					
5500	024737	020							
5501	024740					65\$:			
5502	024740					ENDTST			
5503	024740					L10077:			
5504	024740	104401				TRAP	C\$ETST		


```
5505 024742 BADHEAD
5506 :***** TEST 31 *****
5507 :*FORCE POWER FAIL TEST
5508 :*SET FORCE POWER FAIL BIT VERIFY THAT PROCESSOR TRAPS TO 24
5509 :*GOING DOWN AND COMING UP. VERIFY ALSO THAT BUS INIT WAS
5510 :*BLOCKED FROM GETTING TO THE M8200,4,6,7 DURING THE POWER FAIL
5511 :*****
5512 :*>>> THE DEFAULT CONDITION IS TO SKIP THIS TEST <<<
5513 :*>>> RUN IT ONLY IF POWERFAIL TRAPS TO 24 RATHER <<<
5514 :*>>> THAN THE BOOTSTRAP MODULE. <<<
5515 024742 BADHEAD
5516 :***** TEST 31 *****
5517
5518 024742 BGNTST
5519 024742 T31::
5520 024742
5521 MYINT
5522 024746 MSTCLR ;R1 CONTAINS BASE M8200,4,6,7 ADDRESS
5523 024752 005737 002432 TST SKIP33 ;MASTER CLEAR M8200,4,6,7
5524 024756 001002 BNE 55$ ; EXIT TEST?
5525 024760 000137 025342 JMP 6$ ; SKIP IF NO - RUN IT
5526 024764 005037 002402 55$: CLR TEMP ; SKIP TEST
5527 024770 013737 000024 002406 MOV @#24,$TMP0 ;PREPARE FOR DELAY
5528 024776 013746 000024 MOV @#24,-(SP) ;SAVE POWER FAIL ADDRESS
5529 025002 012737 025064 000024 MOV #1$,@#24 ;STORE POWER FAIL ADDRESS
5530 025010 012761 000002 000004 MOV #2,4(R1) ;SET U FOPR FORCE POWER FAIL
5531 025016 012711 001000 MOV #BIT9,(R1) ;LOAD PORT4
5532 025022 012761 121111 000006 MOV #121111,6(R1) ;SET ROMI
5533 025030 012711 005400 MOV #BIT9!BIT8!BIT11,(R1) ;LOAD INSTRUCTION
5534 025034 005237 002402 5$: INC TEMP ;CLOCK INSTRUCTION
5535 025040 001375 BNE 5$ ;WAIT FOR POWER FAIL
5536 025042 MSTCLR ;BR IF DELAY NOT DONE
5537 025046 ERROR 17 ;ERROR, NO POWER FAIL
5538 025052 104455 TRAP C$ERDF
5539 025054 000021 .WORD 17
5540 025056 005425 .WORD EMO
5541 025060 011240 .WORD ERR17
5542 025062 000445 BR 4$
5543 025064 012737 025102 000024 1$: MOV #3$,@#24 ;POWER UP ADDRESS
5544 025072 010637 025100 MOV SP,2$ ;STORE STACK
5545 025076 000000 HALT ;WAIT FOR POWER UP SEQUENCE
5546 025100 000000 2$: 0
5547 025102 013706 025100 3$: MOV 2$,SP ;RESTORE STACK
5548 025106 012737 025302 000024 MOV #10$,@#24 ;PUT IN CASE OF FALSE POWER-UP.
5549 025114 005037 025300 CLR 11$
5550 025120 005237 025300 12$: INC 11$ ;STALL ON POWER UP.
5551 025124 001375 BNE 12$ ;WAIT HERE IF BAD,WILL POWER OUT OF HERE.
5552 :ELSE PROCEED.
5553
5554 025126 POPSP2 ;POP STACK TWICE2
5555 025130 013701 002456 MOV KMCSR,R1
5556 025134 012637 000024 MOV (SP)+,@#24 ;RESTORE TRUE POWER FAIL ADDRESS
5557 025140 023737 002406 000024 CMP $TMP0,@#24 ;IS IT CORRECT?
5558 025146 001413 BEQ 4$ ;BR IF YES
5559 025150 ERROR 17 ;ERROR, STACK IS INCORRECT
5560 025154 104455 TRAP C$ERDF
```

```

5561 025156 000021 .WORD 17
5562 025160 005425 .WORD EMO
5563 025162 011240 .WORD ERR17
5564 025164 013737 002406 000024 MOV $TMP0,@#24 ;RESTORE TRUE POWER FAIL ADDRESS
5565 025172 013706 002306 MOV PSTACK,SP ;RESTORE STACK
5566 025176 032711 004000 4$: BIT #BIT11,(R1) ;BIT11 STILL SET?
5567 025202 001014 BNE 7$
5568 025204 011104 MOV (R1),R4
5569 025206 012737 004000 002414 MOV #BIT11,$GDDAT
5570 025214 ERROR 35 ;OAC FAILED
5571 025220 104455 TRAP C$ERDF
5572 025222 000043 .WORD 35
5573 025224 005425 .WORD EMO
5574 025226 011534 .WORD ERR35
5575
5576 ;TO PREVENT
5577 ;INIT FROM
5578 ;CLEARING CSR
5578 025230 EXIT TST
5579 025230 104432 TRAP C$EXIT
5580 025232 000110 .WORD L10100-
5581 025234 012711 003000 002414 7$: MOV #BIT9!BIT10,(R1) ;SEL6 = MAINT IR
5582 025240 012737 121111 MOV #121111,$GDDAT ;$GDDAT = EXPECTED
5583 025246 016104 000006 MOV 6(R1),R4 ;R4 = FOUND
5584 025252 023704 002414 CMP $GDDAT,R4 ;MAINT IR SHOULD = 12111
5585 025256 001431 BEQ 6$ ;BR IF OK
5586 025260 MSTCLR
5587 025264 ERROR 35 ;IF = 0 THEN BUS INIT WAS
5588 025270 104455 TRAP C$ERDF
5589 025272 000043 .WORD 35
5590 025274 005425 .WORD EMO
5591 025276 011534 .WORD ERR35
5592
5593 ;NOT BLOCKED FROM CLEARING
5594 ;THE M8200,4,6,7
5595 025300 000000 11$: .WORD 0 ;TEMP COUNT FOR STALL ON POWER UP.
5596
5597 025302 052711 040000 10$: BIS #BIT14,(R1) ;CLR THE THING SO IT CAN'T ASSIRT AC LOW
5598 ;AGAIN!
5599 MSTCLR
5600 ERROR 17 ;ERROR GLIP GAVE US SECOUND UNEXPECTED
5601 025316 104455 TRAP C$ERDF
5602 025320 000021 .WORD 17
5603 025322 005425 .WORD EMO
5604 025324 011240 .WORD ERR17
5605
5606 ;ASSERTION OF AC LOW ON UNIBUS.
5607 ;FATEL TYPE OF ERROR.
5607 025326 062706 000004 ADD #4,SP ;RESTORE STACK.
5608 025332 012637 000024 MOV (SP)+,@#24
5609 025336 MSTCLR
5610 025342 6$:
5611 025342 ENDTST
5612 025342 L10100:
5613 025342 104401 TRAP C$ETST
  
```



```
5614 025344 BADHEAD
5615 :*****TEST 32*****
5616 :*MICRO-PROCESSOR NOISE TEST
5617 :*WRITE ALL ZERO'S THEN ALL ONE'S THEN A DATA PATTERN
5618 :*TO THE IBUS* AND IBUS REGISTERS AND TO THE SP AND MAIN MEM
5619 :*THEN GO BACK AND READ THE DATA PATTERNS TO VERIFY THAT
5620 :*READING AND WRITING OF OTHER LOCATIONS AND REGISTERS
5621 :*DID NOT CHANGE THE DATA.
5622 025344 BADHEAD
5623 :*****TEST 32*****
5624
5625 025344 BGNTST
5626 025344 T32::
5627 025344
5628 025350 MYINT
5629 025354 005002 MSTCLR :MASTER CLEAR M8200,4,6,7
5630 025356 042737 000017 025404 1$: CLR R2 :R2 IS INDEX REGISTER
5631 025364 156237 026374 025404 BIC #17,2$ :CLEAR ADDRESS FIELD
5632 025372 116261 026402 000004 BISB 30$(R2),2$ :ADD IBUS* REG ADDRESS TO INSTRUCTION
5633 025400 MOVB 31$(R2),4(R1) :LOAD PORT4
5634 025404 121100 ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5635 025406 005202 2$: 121100 :WRITE IBUS* REGISTER
5636 025410 022702 000005 INC R2 :INC INDEX REGISTER
5637 025414 001360 CMP #5,R2 :DONE YET?
5638 025416 005002 BNE 1$ :BR IF NO
5639 025420 042737 000017 025466 3$: CLR R2 :R2 IS IBUS REGISTER ADDRESS
5640 025426 042737 000017 025502 BIC #17,4$ :CLEAR ADDRESS FIELD OF INSTRUCTIONS
5641 025434 042737 000017 025514 BIC #17,5$
5642 025442 050237 025466 BIS #17,6$
5643 025446 050237 025502 BIS R2,4$ :ADD IBUS REG ADDRESS TO INSTRUCTION
5644 025452 050237 025514 BIS R2,5$
5645 025456 105061 000004 BIS R2,6$
5646 025462 CLR B 4(R1) :CLEAR PORT4
5647 025466 122100 ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5648 025470 112761 000377 000004 4$: 122100 :WRITE 0 TO IBUS REG
5649 025476 MOV B #377,4(R1) :LOAD PORT4
5650 025502 122100 ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5651 025504 110261 000004 5$: 122100 :WRITE ALL ONES TO IBUS REG
5652 025510 MOV B R2,4(R1) :LOAD PORT4
5653 025514 122100 ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5654 025516 005202 6$: 122100 :WRITE ITS OWN ADDRESS TO IBUS REG
5655 025520 022702 000010 INC R2 :NEXT ADDRESS
5656 025524 001335 CMP #10,R2 :DONE YET?
5657 025526 005002 BNE 3$ :BR IF NO
5658 025530 042737 000017 025576 7$: CLR R2 :START AT SP ADDRESS 0
5659 025536 042737 000017 025612 BIC #17,8$ :CLEAR ADDRESS FIELD
5660 025544 042737 000017 025624 BIC #17,9$
5661 025552 050237 025576 BIS #17,10$
5662 025556 050237 025612 BIS R2,8$ :ADD ADDRESS TO INSTRUCTION
5663 025562 050237 025624 BIS R2,9$
5664 025566 105061 000004 BIS R2,10$
5665 025572 CLR B 4(R1) :CLEAR PORT4
5666 025576 123100 ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5667 025600 112761 000377 000004 8$: 123100 :WRITE ZERO TO SP
5668 025606 MOV B #377,4(R1) :LOAD PORT4
5669 025612 123100 ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
:WRITE ALL ONES TO SP
```



```

5670 025614 110261 000004      MOVB      R2,4(R1)      :LOAD PORT4
5671 025620                    ROMCLK          :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5672 025624 123100      10$:      123100      :WRITE SP ADDRESS TO ITSELF
5673 025626 005202      INC          R2      :NEXT SP ADDRESS
5674 025630 022702 000020      CMP      #20,R2      :DONE YET?
5675 025634 001335      BNE      7$          :BR IF NO
5676 025636 005002      CLR          R2      :R2 = ,AOM ,E, ADDRESS
5677 025640                    ROMCLK          :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5678 025644 010000      010000      :MAR _ 0
5679 025646                    ROMCLK          :
5680 025652 004000      4000
5681 025654 105061 000004      11$:      CLRB      4(R1)      :CLEAR PORT4
5682 025660                    ROMCLK          :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5683 025664 122500      122500      :WRITE ZEROS TO MEM
5684 025666 112761 000377 000004      MOVB      #377,4(R1) :LOAD PORT4
5685 025674                    ROMCLK          :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5686 025700 122500      122500      :WRITE ONES TO MEM
5687 025702 110261 000004      MOVB      R2,4(R1)   :LOAD PORT4
5688 025706                    ROMCLK          :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5689 025712 136500      136500      :WRITE TO MEM IT OWN ADDRESS
5690 025714 005202      INC          R2      :NEXT MEM ADDRESS
5691 025716 022702 001000      CMP      #1000,R2   :DONE YET?
5692 025722 001354      BNE      11$        :BR IF NO
5693
5694      :NOW GO BACK AND READ EVERYTHIN
5695
5696 025724                    ROMCLK          :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5697 025730 010000      010000      :MAR 0
5698 025732                    ROMCLK          :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5699 025736 004000      4000      :MAR HI _ 0 (M8200,4,6,7 ONLY)
5700      :WOULD BE CRAM CODE
5701 025740 005737 002374      TST      TYPE
5702 025744 001446      BEQ      40$
5703 025746 005005      CLR      R5
5704 025750 042737 000360 026012 12$:      BIC      #360,13$   :R5 IS INDEX REGISTER
5705 025756 116502 026374      MOVB     30$(R5),R2 :CLEAR ADDRESS FIELD
5706 025762 010203      MOV      R2,R3      :R2 = IBUS* ADDRESS
5707 025764 006303      ASL     R3          :PUT IBUS* ADDRESS IN R3
5708 025766 006303      ASL     R3          :SHIFT ADDRESS TO BITS 4-7
5709 025770 006303      ASL     R3
5710 025772 006303      ASL     R3
5711 025774 050337 026012      BIS     R3,13$      :ADD ADDRESS TO INSTRUCTION
5712 026000 116537 026402 002414      MOVB     31$(R5),$GDDAT :$GDDAT = 'EXPECTED'
5713 026006                    ROMCLK          :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5714 026012 121004      13$:      121004      :PORT4 - IBUS* REGISTER
5715 026014 016104 000004      MOV      4(R1),R4   :R4 = 'FOUND'
5716 026020 123704 002414      CMPB     $GDDAT,R4  :IBUS* CONTENTS OK?
5717 026024 001412      BEQ     20$        :BR IF YES
5718 026026 010237 002376      MOV      R2,MRO
5719 026032 105037 002415      CLRB     $GDDAT+1
5720 026036      ERROR    29
5721 026042 104455      TRAP     C$ERDF
5722 026044 000035      .WORD   29
5723 026046 005425      .WORD   EMO
5724 026050 011362      .WORD   ERR29
5725 026052 005205      20$:      INC      R5          :INC COUNTER

```

5726	026054	022705	000005				CMP #5,R5	:DONE YET?
5727	026060	001333					BNE 12\$:BR IF NO
5728								
5729	026062				40\$:			
5730								:END CRAM,GENERAL TESTS
5731								
5732	026062	005002					CLR R2	:R2 = IBUS REG ADDRESS
5733	026064	042737	000360	026120	14\$:		BIC #360,15\$:CLEAR ADDRESS FIELD OF INSTRUCTION
5734	026072	010203					MOV R2,R3	:R3 = IBUS ADDRESS
5735	026074	006303					ASL R3	:SHIFT ADDRESS TO BITS 4-7
5736	026076	006303					ASL R3	
5737	026100	006303					ASL R3	
5738	026102	006303					ASL R3	
5739	026104	050337	026120				BIS R3,15\$:ADD ADDRESS TO INSTRUCTION
5740	026110	010237	002414				MOV R2,\$GDDAT	:\$GDDAT = 'EXPECTED'
5741	026114						ROMCLK	:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5742	026120	021004			15\$:		021004	:PORT4 - IBUS REG
5743	026122	016104	000004				MOV 4(R1),R4	:IBUS = 'FOUND'
5744	026126	123704	002414				CMPB \$GDDAT,R4	:IBUS CONTENTS OK?
5745	026132	001406					BEQ 21\$:BR IF YES
5746	026134						ERROR 29	:IBUS DATA ERROR
5747	026140	104455					TRAP C\$ERDF	
5748	026142	000035					.WORD 29	
5749	026144	005425					.WORD EMO	
5750	026146	011362					.WORD ERR29	
5751	026150	005202			21\$:		INC R2	:NEXT IBUS REGISTER
5752	026152	022702	000010				CMP #10,R2	:DONE YET?
5753	026156	001342					BNE 14\$:BR IF NO
5754	026160	005002					CLR R2	:R2 = SP ADDRESS
5755	026162	042737	000017	026200	16\$:		BIC #17,17\$:CLEAR ADDRESS FIELD OF INSTRUCTION
5756	026170	050237	026200				BIS R2,17\$:ADD ADDRESS TO INSTRUCTION
5757	026174						ROMCLK	:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5758	026200	040600			17\$:		040600	:BR - SP
5759	026202	010237	002414				MOV R2,\$GDDAT	:\$GDDAT = 'EXPECTED'
5760	026206						ROMCLK	:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5761	026212	061224					061224	:PORT4 BR
5762	026214	016104	000004				MOV 4(R1),R4	:R4 = 'FOUND'
5763	026220	123704	002414				CMPB \$GDDAT,R4	:SP CONTENTS OK?
5764	026224	001406					BEQ 22\$:BR IF YES
5765	026226						ERROR 7	:SP DATA ERROR
5766	026232	104455					TRAP C\$ERDF	
5767	026234	000007					.WORD 7	
5768	026236	005425					.WORD EMO	
5769	026240	007510					.WORD ERR7	
5770	026242	005202			22\$:		INC R2	:NEXT SP LOCATION
5771	026244	022702	000020				CMP #20,R2	:DONE YET?
5772	026250	001344					BNE 16\$:BR IF NO
5773	026252	005002					CLR R2	:R2 = MEMORY ADDRESS
5774	026254						ROMCLK	:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5775	026260	010000					010000	:MAR 0
5776	026262						ROMCLK	:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5777	026266	004000					4000	:MAR HI 0 (M8200,4,6,7 OR FAMILY ONLY)
5778	026270	010237	002414		18\$:		MOV R2,\$GDDAT	:\$GDDAT = 'EXPECTED'
5779	026274						ROMCLK	:NEXT WORD IS INSTRUCTION, ROMCLK=5304
5780	026300	055224					055224	:PORT4 MAIN MEM
5781	026302	016104	000004				MOV 4(R1),R4	:R4 = 'FOUND'

5782	026306	123704	002414			CMPB	\$GDDAT,R4		:MAIN MEM CONTENTS OK?
5783	026312	001406				BEQ	23\$:BR IF YES
5784	026314					ERROR	6		:MAIN MEM DATA ERROR
5785	026320	104455				TRAP	C\$ERDF		
5786	026322	000006				.WORD	6		
5787	026324	005425				.WORD	EMO		
5788	026326	007332				.WORD	ERR6		
5789	026330	005202		23\$:		INC	R2		:NEXT MEM ADDRESS
5790	026332	022702	001000			CMP	#1000,R2		:DONE YET?
5791	026336	001354				BNE	18\$:BR IF NO
5792	026340					CHKREG			
5793						.LIST	ME		
5794	026340	004737	003014			JSR	PC,RDKM		: READ KMC11 REGISTERS
5795	026344	012702	026356			MOV	#64\$,R2		: LOAD POINTER
5796	026350	004737	003120			JSR	PC,RCHECK		: CHECK FOR NO CHANGE
5797	026354	000405				BR	65\$: SKIP LIST
5798	026356	000	000	000	64\$:	.BYTE20		
5799	026361	000	000	000					
5800	026364	000	000	000					
5801	026367	020							
5802	026370				65\$:				
5803	026370					EXIT	TST		
5804	026370	104432				TRAP	C\$EXIT		
5805	026372	000046				.WORD	L10101-		
5806	026374	000	002	003	30\$:	.BYTE	0,2,3,5,10		
5807	026377	005	010						
5808									
5809		026402			.EVEN				
5810	026402	001	003	004	31\$:	.BYTE	1,3,4,6,10		
5811	026405	006	010						


```
5812
5813          026410          .EVEN
5814
5815 026410          CHKREG
5816          .LIST      ME
5817 026410 004737 003014  JSR      PC,RDKM      : READ KMC11 REGISTERS
5818 026414 012702 026426  MOV      #66$,R2      : LOAD POINTER
5819 026420 004737 003120  JSR      PC,RCHECK    : CHECK FOR NO CHANGE
5820 026424 000405          BR      67$           : SKIP LIST
5821 026426          000      000      000 66$: .BYTE .....20
5822 026431          000      000      000
5823 026434          000      000      000
5824 026437          020
5825 026440          67$:
5826 026440          ENDTST
5827 026440          L10101:
5828 026440 104401          TRAP      CSETST
```

```
5829 026442 BADHEAD
5830 :***** TEST 33 *****
5831 :* THIS TEST IS DESIGNED TO MAKE SURE THAT A NODST INSTRUCTION
5832 :* DOES NOT WRITE INTO PORT B OF THE MULTIPOINT RAM.
5833 :* TO DO THIS,WE'LL PUT A 125 INTO INDAT2,THEN WE'LL PUT A
5834 :* 125 INTO BOTH SP1 AND BR. LAST WE'LL DO A NODST BR,SUBOC,SP1
5835 :* IF THERE IS A WRITE INTO PORTB,INDAT2 WILL CONTAIN A 377.
5836 026442 BADHEAD
5837 :***** TEST 33 *****
5838
5839 026442 BGNTST
5840 026442 T33::
5841 026442 MYINT
5842 026446 MSTCLR
5843 026452 ROMCLK
5844 026456 000525 00525 ;PUT A 125 INTO BRG.
5845 026460 ROMCLK
5846 026464 062221 062221 ;NOW INTO OIDAT2
5847 026466 ROMCLK
5848 026472 063221 63221 ;NOW INTO SP1
5849 026474 ROMCLK
5850 026500 060361 060361 ;NOW THE 'NODST BR,SUBOC,SP1'
5851 ;THE NODST SHOULD NOT MODIFY INDAT2!
5852
5853 026502 ROMCLK
5854 026506 020420 020420 ;PUT CONTENT OF INDAT2 IN BRG.
5855
5856 026510 ROMCLK
5857 026514 061220 061220 ;PUT BRG INTO BSELO
5858
5859 026516 111137 002416 MOV (R1),SBDDAT ;SEE WHAT CAME BACK.
5860 026522 012737 000125 002414 MOV #125,$GDDAT ;SHOULD BE 125 IF 377 CAME BACK.
5861 026530 105037 002417 CLR SBDDAT+1 ;YOU CAN BET THAT THE 'NODST' WROTE
5862 ;INTO THE MULTIPOINT RAM! WATCH SIGNAL
5863 ; 'D1 WRITE OUT L'
5864
5865 026534 023737 002416 002414 CMP SBDDAT,$GDDAT ;NOW LOOK.
5866 026542 001406 BEQ 10$
5867
5868 026544 ERROR 7
5869 026550 104455 TRAP C$ERDF
5870 026552 000007 .WORD 7
5871 026554 005425 .WORD EMO
5872 026556 007510 .WORD ERR7
5873
5874 026560 10$:
5875 026560 CHKREG
5876 .LIST ME
5877 026560 004737 003014 JSR PC,RDKM ; READ KMC11 REGISTERS
5878 026564 012702 026576 MOV #64$,R2 ; LOAD POINTER
5879 026570 004737 003120 JSR PC,R$CHECK ; CHECK FOR NO CHANGE
5880 026574 000405 BR 65$ ; SKIP LIST
5881 026576 000 000 000 64$: .BYTE .....20
5882 026601 000 000 000
5883 026604 000 000 000
5884 026607 020
```

CZKMCAD KMC11-B STATIC PART2
CZKMC.A.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 ⁸12 PAGE 144
HARDWARE TESTS

SEQ 0144

5885 026610
5886 026610
5887 026610
5888 026610 104401

65\$:
L10102: ENDTST
TRAP CSETST


```

5889 026612          BADHEAD
5890                :***** TEST 34 *****
5891                :*
5892                :* EXTENDED CRAM TEST FOR M8206. IN THIS TEST WE WILL LOAD DATA
5893                :* THROUGHOUT THE CRAM (TEST DATA IS JUST 4K OF DIAG. CODE) AND
5894                :* THEN READ IT BACK AND VERIFY THAT IT IS CORRECT.
5895 026612          BADHEAD
5896                :***** TEST 34 *****
5897
5898 026612          BGNTST
5899 026612          T34::
5900
5901 026612          10$: MYINT
5902 026616          MSTCLR
5903 026622 012702 013116  MOV      #ROMMAP,R2          ;GET ADDR. OF LIST.
5904
5905 026626 012711 002000  MOV      #2000,(R1)         ;SET TO WRITE DATA.
5906 026632 005003          CLR      R3                  ;CRAM ADDR ZERO.
5907
5908 026634 010361 000004  15$: MOV      R3,4(R1)         ;SET ADDR.
5909 026640 012261 000006  MOV      (R2)+,6(R1)       ;WRITE DATA.
5910
5911 026644 020337 002400  CMP      R3,MEMSZ         ;DONE WHOLE CRAM?
5912 026650 001402          BEQ      20$              ;YES,EXIT THIS LOOP.
5913 026652 005203          INC      R3                  ;NO,UPDAT ADDR.
5914 026654 000767          BR       15$
5915 026656 005003          20$: CLR      R3                  ;NOW WE WILL READ BACK,STARTING AT
5916                ;CRAM ADDR. ZERO.
5917 026660 012705 013116  MOV      #ROMMAP,R5       ;GET ADDR. LIST OF DATA
5918
5919 026664 010361 000004  30$: MOV      R3,4(R1)         ;SET ADDR.
5920
5921 026670 011502          MOV      (R5),R2          ;PUT EXPECTED INTO R2
5922 026672 016104 000006  MOV      6(R1),R4         ;READ ACCUAL
5923 026676 020204          CMP      R2,R4           ;EQUAL?
5924 026700 001412          BEQ      40$              ;YES,CONTINUE.
5925 026702 010337 002376  MOV      R3,MRO
5926
5927 026706          ERROR      1          ;ERROR CRAM DATA TEST,DATA
5928 026712 104455          TRAP     C$ERDF
5929 026714 000001          .WORD   1
5930 026716 005425          .WORD   EMO
5931 026720 006274          .WORD   ERR1
5932
5933                ;READ NOT DATA THAT WAS WRITTEN.
5934 026722          ESCAPE TST
5935 026722 104410          TRAP     C$ESCAPE
5936 026724 000054          .WORD   L10103-
5937 026726 020337 002400  40$: CMP      R3,MEMSZ         ;ALL DONE?
5938 026732 001016          BNE
5939 026734          CHKREG
5940                .LIST
5941 026734 004737 003014  JSR      PC,RDKM          ; READ KMC11 REGISTERS
5942 026740 012702 026752  MOV      #64$,R2         ; LOAD POINTER
5943 026744 004737 003120  JSR      PC,R$CHECK      ; CHECK FOR NO CHANGE
5944 026750 000405          BR       65$             ; SKIP LIST

```

5945	026752	000	000	000	64\$:	.BYTE,20	
5946	026755	000	000	000				
5947	026760	000	000	000				
5948	026763	020						
5949	026764				65\$:			
5950								
5951	026764					EXIT	TST	
5952	026764	104432				TRAP	C\$EXIT	
5953	026766	000012				.WORD	L10103-	
5954								
5955	026770	005203			50\$:	INC	R3	:UPDATE ADDR.
5956	026772	062705	000002			ADD	#2,R5	
5957	026776	000732				BR	30\$	
5958								
5959	027000					ENDTST		
5960	027000					L10103:		
5961	027000	104401				TRAP	C\$ETST	

```
5962 027002          BADHEAD
5963                :***** TEST 35 *****
5964                :*
5965                :* THIS TEST LOADS MICRO-CODE INTO A M8206 MCPU THEN EXECUTES IT.
5966                :* THE MICRO CODE IS DESIGNED TO WRITE ALL ONES INTO THE SEL REGS.
5967                :*
5968 027002          BADHEAD
5969                :***** TEST 35 *****
5970
5971 027002          BGNTST
5972 027002          T35::
5973
5974
5975 027002          1$: MYINT
5976 027006          MSTCLR
5977
5978 027012 004537 027446 JSR      R5,LOADER          ;LOAD THE MICRO CODE
5979
5980 027016 000777          777          ;MOVE #377,BRG
5981 027020 061220          061220        ;MOVE BRG,BSEL0
5982 027022 061222          061222        ;MOVE BRG,BSEL2
5983 027024 061223          061223        ;MOVE BRG,BSEL3
5984 027026 061224          061224        ;MOVE BRG,BSEL4
5985 027030 061225          061225        ;MOVE BRG,BSEL5
5986 027032 061226          061226        ;MOVE BRG,BSEL6
5987 027034 061227          061227        ;MOVE BRG,BSEL7
5988 027036 123000          123000        ;MOVE BSEL0,SPO
5989 027040 101410          101410        ;BRANCH BACK ONE UNTIL <>377
5990
5991 027042 000400          400          ;MOVE #0,BRG
5992 027044 061220          61220        ;MOVE BRG,BSEL0
5993 027046 061222          61222        ;MOVE BRG,BSEL2
5994 027050 061223          61223        ;MOVE BRG,BSEL3
5995 027052 061224          61224        ;MOVE BRG,BSEL4
5996 027054 061225          61225        ;MOVE BRG,BSEL5
5997 027056 061226          61226        ;MOVE BRG,BSEL6
5998 027060 061227          61227        ;MOVE BRG,BSEL7
5999 027062 123000          123000        ;MOVE BSEL0,SPO
6000 027064 104022          104022        ;BRANCH BACK ONE LOCATION.
6001 027066 177777          177777
6002
6003 027070 005011          CLR      @R1          ; CLEAR RUN
6004 027072 012711 040000  MOV      #040000,(R1) ; INITIALIZE MCPU
6005 027076 012711 100000  MOV      #100000,(R1) ; START CPU.
6006
6007 027102 012700 000062  MOV      #50.,R0     ; THE CYCLE TIME ON THE M8206 IS
6008                :200NS. WE ARE ASKING THE MCPU TO
6009                :DO 8 INSTRUCTIONS. WE'LL DELAY
6010                :100 PDP11 INSTRUCTIONS
6011                :THIS REALLY SHOULD BE PLENTY OF TIME.
6012
6013 027106 005300          20$: DEC      R0
6014 027110 001376          BNE     20$
6015
6016 027112 005005          CLR      R5
6017 027114 012705 000377  MOV      #377,R5     ; JUST FOR TYPEOUT.
                        :EXPECT 377
```


6018	027120	111104			MOVB	(R1),R4		:READ MCPU
6019	027122	120405			CMPB	R4,R5		:SEE IF OK.
6020	027124	001410			BEQ	30\$		
6021								
6022	027126				ERROR	29		:ERROR! MCPU WAS TO WRITE ALL
6023	027132	104455			TRAP	C\$ERDF		
6024	027134	000035			.WORD	29		
6025	027136	005425			.WORD	EMO		
6026	027140	011362			.WORD	ERR29		
6027								:ONES INTO BSELO,BUT INSTEAD FAILED.
6028	027142				ESCAPE	TST		
6029	027142	104410			TRAP	C\$ESCAPE		
6030	027144	000334			.WORD	L10104-		
6031								
6032	027146	012705	177777	30\$:	MOV	#177777,R5		:EXPECT ALL ONES
6033	027152	016104	000002		MOV	2(R1),R4		:RECIEVED
6034	027156	020405			CMP	R4,R5		:RECIEVE OK?
6035	027160	001410			BEQ	40\$		
6036								
6037	027162				ERROR	29		:ERROR! MCPU WAS TO WRITE ALL ONES
6038	027166	104455			TRAP	C\$ERDF		
6039	027170	000035			.WORD	29		
6040	027172	005425			.WORD	EMO		
6041	027174	011362			.WORD	ERR29		
6042								:INTO BSEL 2&3
6043								
6044	027176				ESCAPE	TST		
6045	027176	104410			TRAP	C\$ESCAPE		
6046	027200	000300			.WORD	L10104-		
6047								
6048	027202	016104	000004	40\$:	MOV	4(R1),R4		:READ BSEL 4&5
6049	027206	020405			CMP	R4,R5		:READ OK?
6050	027210	001410			BEQ	50\$		
6051								
6052	027212				ERROR	29		:ERROR! FAILED TO WRITE BSEL \$&5
6053	027216	104455			TRAP	C\$ERDF		
6054	027220	000035			.WORD	29		
6055	027222	005425			.WORD	EMO		
6056	027224	011362			.WORD	ERR29		
6057								: TO ALL ONES.
6058	027226				ESCAPE	TST		
6059	027226	104410			TRAP	C\$ESCAPE		
6060	027230	000250			.WORD	L10104-		
6061								
6062	027232	016104	000006	50\$:	MOV	6(R1),R4		:READ BSEL 6&7
6063	027236	020405			CMP	R4,R5		:READ OK?
6064	027240	001410			BEQ	60\$		
6065								
6066	027242				ERROR	29		:ERROR! FAILED TO WRITE BSEL 6&7
6067	027246	104455			TRAP	C\$ERDF		
6068	027250	000035			.WORD	29		
6069	027252	005425			.WORD	EMO		
6070	027254	011362			.WORD	ERR29		
6071								: TO ALL ONES.
6072	027256				ESCAPE	TST		
6073	027256	104410			TRAP	C\$ESCAPE		

6074	027260	000220			.WORD	L10104-		
6075	027262	105011		60\$:	CLRB	(R1)		:SIGNAL MCPU TO WRITE ALL ZEROS.
6076	027264	005005			CLR	R5		:EXPECT TO READ ALL ZEROS.
6077								
6078	027266	005004			CLR	R4		
6079	027270	111104			MOVB	(R1),R4		:READ BSELO
6080	027272	001410			BEQ	70\$:EXPECT ZERO.
6081								
6082	027274				ERROR	29		:MCPU FAILED TO CLEAR BSELO
6083	027300	104455			TRAP	C\$ERDF		
6084	027302	000035			.WORD	29		
6085	027304	005425			.WORD	EMO		
6086	027306	011362			.WORD	ERR29		
6087								
6088	027310				ESCAPE	TST		
6089	027310	104410			TRAP	C\$ESCAPE		
6090	027312	000166			.WORD	L10104-		
6091	027314	016104	000002	70\$:	MOV	2(R1),R4		:READ BSEL 2&3
6092	027320	001410			BEQ	80\$:IF ZERO,OK
6093								
6094	027322				ERROR	29		:MCPU FAILED TO CLEAR BSEL 2&3
6095	027326	104455			TRAP	C\$ERDF		
6096	027330	000035			.WORD	29		
6097	027332	005425			.WORD	EMO		
6098	027334	011362			.WORD	ERR29		
6099	027336				ESCAPE	TST		
6100	027336	104410			TRAP	C\$ESCAPE		
6101	027340	000140			.WORD	L10104-		
6102	027342			80\$:				
6103	027342	016104	000004		MOV	4(R1),R4		:READ BSEL 4&5
6104	027346	001410			BEQ	90\$		
6105								
6106	027350				ERROR	29		:MCPU FAILED TO CLEAR BSEL 4&5
6107	027354	104455			TRAP	C\$ERDF		
6108	027356	000035			.WORD	29		
6109	027360	005425			.WORD	EMO		
6110	027362	011362			.WORD	ERR29		
6111	027364				ESCAPE	TST		
6112	027364	104410			TRAP	C\$ESCAPE		
6113	027366	000112			.WORD	L10104-		
6114	027370			90\$:				
6115	027370	016104	000006		MOV	6(R1),R4		:READ BSEL 6&7
6116	027374	001406			BEQ	95\$		
6117								
6118	027376				ERROR	29		:MCPU FAILED TO CLEAR BSEL 6&7
6119	027402	104455			TRAP	C\$ERDF		
6120	027404	000035			.WORD	29		
6121	027406	005425			.WORD	EMO		
6122	027410	011362			.WORD	ERR29		
6123								
6124	027412			95\$:				
6125	027412				CHKREG			
6126					.LIST	ME		
6127	027412	004737	003014		JSR	PC,RDKM		: READ KMC11 REGISTERS
6128	027416	012702	027430		MOV	#64\$,R2		: LOAD POINTER
6129	027422	004737	003120		JSR	PC,R\$CHECK		: CHECK FOR NO CHANGE

```
6130 027426 000405  
6131 027430 000 000 000 64$: BR 65$ ; SKIP LIST  
6132 027433 000 000 000  
6133 027436 000 000 000  
6134 027441 020  
6135 027442 65$:  
6136 027442  
6137 027442 104432 EXIT TST  
6138 027444 000034 TRAP C$EXIT  
6139 .WORD L10104-  
6140  
6141  
6142 ;:LOADER SUBROUTINE USED BY THIS TEST TO LOAD MICRO CODE INTO A M8206  
6143 ;  
6144  
6145 027446 012711 002000 LOADER: MOV #2000,(R1)  
6146  
6147 027452 005000 CLR R0  
6148  
6149 027454 010061 000004 10$: MOV R0,4(R1) ;SET ADDR.  
6150 027460 005200 INC R0  
6151 027462 011561 000006 MOV (R5),6(R1) ;WRITE MICRO CODE.  
6152 027466 022527 177777 CMP (R5)+,#177777 ;SEE IF TERM.  
6153 027472 001370 BNE 10$  
6154 027474 005011 CLR (R1)  
6155 027476 000205 RTS R5  
6156  
6157 027500  
6158 027500 L10104: ENDTST  
6159 027500 104401 TRAP C$ETST
```



```

6160 027502          BADHEAD
6161                :***** TEST 36 *****
6162                :*
6163                :*NEGATIVE ADDRESS TEST.
6164                :*      IN THIS TEST, WE'LL MAKE SURE THAT THE M8206
6165                :*      DOES NOT RESPOND TO AN ADDRESS THAT ISN'T ASSIGNED
6166                :*      TO IT
6167                :*
6168 027502          BADHEAD
6169                :***** TEST 36 *****
6170
6171 027502          BGNTST
6172 027502          T36::
6173 027502          MYINT
6174 027506          MSTCLR
6175
6176 027512 012711 000641          MOV      #641,(R1)          ;PUT A DEFINITE PATTERN IN MCPU.
6177 027516 012737 027624 000004  MOV      #20$,a#4          ;SET UP FOR TRAPS FROM NON-EX.
6178 027524 005037 000006          CLR      a#6
6179 027530 012702 160000          MOV      #160000,R2        ;GET STARTING ADDRESS.
6180
6181 027534 022712 000641          10$:    CMP      #641,(R2)        ;SEE IF CONTENTS OF THE ADDRESS
6182                :POINTED TO BY R2 EQUALS THE CONTENTS
6183                :OF THE MCPU CSR
6184 027540 001434          BEQ      40$
6185
6186 027542 062702 000002          15$:    ADD      #2,R2          ;UPDATE ADDRESS.
6187 027546 020227 177750          CMP      R2,#177750        ;DONE?
6188 027552 001370          BNE      10$              ;NO-LOOP
6189
6190 027554 013737 002426 000004  17$:    MOV      SAVE4,a#4          ;RESTORE TRAP CATCHER
6191 027562 013737 002430 000006  MOV      SAVE6,a#6          ;FROM VALUES SAVED BY INIT SECTION
6192 027570
6193                CHKREG
6194 027570 004737 003014          .LIST  ME
6195 027574 012702 027606          JSR      PC,RDKM           ; READ KMC11 REGISTERS
6196 027600 004737 003120          MOV      #64$,R2          ; LOAD POINTER
6197 027604 000405          JSR      PC,RCHECK        ; CHECK FOR NO CHANGE
6198 027606          000          000          BR      65$              ; SKIP LIST
6199 027611          000          000          64$:    .BYTE  .....20
6200 027614          000          000
6201 027617          020
6202 027620          65$:
6203 027620          EXIT      TST          ;EXIT, ALL DONE
6204 027620 104432          TRAP    CSEXIT
6205 027622 000052          .WORD  L10105-.
6206
6207 027624 062706 000004          20$:    ADD      #4,SP          ;SAVE FROM TRAP
6208 027630 000744          BR      15$              ;LOOP
6209
6210 027632          40$:    ;*OH NO, WE MAY HAVE A DUAL ADDRESS PROBLEM!
6211
6212 027632 012711 000174          MOV      #174,(R1)        ;WRITE NEW PATTERN IN MCPU CSR
6213 027636 022712 000174          CMP      #174,(R2)        ;DID NEW PATTERN SHOW UP IN ADDR?
6214 027642 001403          BEQ      60$
6215

```

6216	027644	012711	000641	50\$:	MOV	#641,(R1)	:PUT OLD PATTERN BACK IN MCPU CSR.
6217	027650	000734			BR	15\$:LOOP
6218							
6219	027652	020102		60\$:	CMP	R1,R2	:IS THIS THE MCPU ADDRESS?
6220	027654	001773			BEQ	50\$:YES-NO ERROR
6221							
6222	027656				ERROR	40	:DUAL ADDRESS ERROR
6223	027662	104455			TRAP	C\$ERDF	
6224	027664	000050			.WORD	40	
6225	027666	005425			.WORD	EMO	
6226	027670	011760			.WORD	ERR40	
6227	027672	000730			BR	17\$	
6228							
6229	027674				ENDTST		
6230	027674			L10105:			
6231	027674	104401			TRAP	C\$ETST	

```
6232 027676 BADHEAD
6233 :***** TEST 37 *****
6234 :*
6235 :*BYTE ADDRESSING TEST
6236 :* HERE, WE'RE GOING TO MAKE SURE THAT WE CAN
6237 :* WRITE INTO ONLY A HIGH OR LOW BYTE OF THE MCPU.
6238 :*
6239 027676 BADHEAD
6240 :***** TEST 37 *****
6241
6242 027676 BGNTST
6243 027676 T37:: MYINT
6244 027676 MSTCLR
6245 027702 CLR (R1) ;CLEAR CSR
6246 027706 005011 177777 MOV#B #-1,(R1) ;WRITE ALL ONES INTO LOW BYTE
6247 027710 112711 177777 ;OF CSR
6248 ;SEE IF HIGH BYTE GOT WRITTEN
6249 027714 032711 177400 BIT #177400,(R1)
6250 027720 001410 BEQ 10$
6251
6252 027722 ERROR 41 ;HIGH BYTE GOT WRITTEN INTO ON A LOW BYTE
6253 027726 104455 TRAP C$ERDF
6254 027730 000051 .WORD 41
6255 027732 005425 .WORD EMO
6256 027734 012024 .WORD ERR41
6257 027736 ESCAPE TST ;OPERATION
6258 027736 104410 TRAP C$ESCAPE
6259 027740 000064 .WORD L10106-.
6260
6261 027742 005011 177777 000001 10$: CLR (R1)
6262 027744 112761 177777 000001 MOV#B #-1,1(R1) ;WRITE INTO HIGH BYTE
6263 027752 032711 000377 BIT #377,(R1) ;SEE IF LOW BYTE GOT WRITTEN
6264 027756 001406 BEQ 20$
6265
6266 027760 ERROR 42 ;LOW BYTE GOT WRITTEN INTO ON A
6267 027764 104455 TRAP C$ERDF
6268 027766 000052 .WORD 42
6269 027770 005425 .WORD EMO
6270 027772 012066 .WORD ERR42
6271
6272 ;HIGH BYTE OPERATION.
6273 027774 20$:
6274 027774 CHKREG
6275 .LIST ME
6276 027774 004737 003014 JSR PC,RDKM ; READ KMC11 REGISTERS
6277 030000 012702 030012 MOV #64$,R2 ; LOAD POINTER
6278 030004 004737 003120 JSR PC,R$CHECK ; CHECK FOR NO CHANGE
6279 030010 000405 BR 65$ ; SKIP LIST
6280 030012 000 000 000 64$: .BYTE .....20
6281 030015 000 000 000
6282 030020 000 000 000
6283 030023 020
6284 030024 65$:
6285 030024 ENDTST
6286 030024 L10106:
6287 030024 104401 TRAP C$ETST
```



```
6288 030026 BADHEAD
6289 :***** TEST 38 *****
6290 :*
6291 :*IN THIS TEST WE'RE GOING TO MAKE SURE THAT THE PC
6292 :*REG COUNTS UP PROPERLY. THE PC REG SHOULD INCREMENT
6293 :*ONCE AFTER EACH INSTRUCTION.
6294 :*
6295 030026 BADHEAD
6296 :***** TEST 38 *****
6297
6298 030026 BGNTST
6299 030026 T38::
6300 030026 10$: MYINT
6301 030032 MSTCLR
6302 030036 005000 CLR R0 : START AT 0
6303 030040 012711 002000 MOV #BIT10,(R1) : SET ROMO
6304 030044 010061 000004 1$: MOV R0,4(R1) : ADDRESS
6305 030050 005061 000006 CLR 6(R1) : LOAD A NOP
6306 030054 005200 INC R0 : NEXT ADDRESS
6307 030056 032700 010000 BIT #BIT12,R0 : DONE 4K?
6308 030062 001770 BEQ 1$ : NO
6309 030064 012705 000000 MOV #0,R5 : START AT ZERO
6310 030070 012737 007777 002354 MOV #7777,FADR : START AT LAST ADDRESS
6311 030076 012711 002000 2$: MOV #BIT10,(R1) : SET ROMO
6312 030102 013761 002354 000004 MOV FADR,4(R1) : SETUP PC
6313 030110 005011 CLR (R1)
6314 030112 012711 000400 MOV #BIT8,(R1) : SET SINGLE STEP
6315 030116 005011 CLR (R1) : RESET SINGLE STEP
6316 030120 20$: ROMCLK : READ PC HIGH REG.
6317 030124 121265 121265
6318
6319 030126 ROMCLK : READ PC LOW REG.
6320 030132 121244 121244
6321 030134 016104 000004 MOV 4(R1),R4 : GET WHOLE PICTURE
6322 030140 042704 170000 BIC #170000,R4
6323 030144 020405 CMP R4,R5 : INCREMENT OK?
6324 030146 001410 BEQ 30$
6325
6326 030150 ERROR 15 : PC FAILED TO INCREMENT PROPERLY
6327 030154 104455 TRAP C$ERDF
6328 030156 000017 .WORD 15
6329 030160 005425 .WORD EMO
6330 030162 010710 .WORD ERR15
6331
6332 : SHOULD INCREMENT BY ONE
6333 : FOR EACH INSTRUCTION.
6334 030164 ESCAPE TST
6335 030166 104410 TRAP C$ESCAPE
6336 030166 000054 .WORD L10107-.
6337 030170 005237 002354 30$: INC FADR : BUMP ADDRESS
6338 030174 042737 010000 002354 BIC #BIT12,FADR : 0-7777
6339 030202 005205 INC R5 : BUMP ADDRESS
6340 030204 032705 010000 BIT #BIT12,R5 : DONE?
6341 030210 001732 BEQ 2$ : NO
6342
6343 030212 CHKREG
```

6344							.LIST	ME	
6345	030212	004737	003014				JSR	PC,RDKM	: READ KMC11 REGISTERS
6346	030216	012702	030230				MOV	#64\$,R2	: LOAD POINTER
6347	030222	004737	003120				JSR	PC,RCHECK	: CHECK FOR NO CHANGE
6348	030226	000405					BR	65\$: SKIP LIST
6349	030230	000	000	000	64\$:		.BYTE20	
6350	030233	000	000	000					
6351	030236	000	000	000					
6352	030241	020							
6353	030242				65\$:				
6354	030242						ENDTST		
6355	030242				L10107:				
6356	030242	104401					TRAP	CSETST	

```
6357 030244 BADHEAD
6358 :***** TEST 39 *****
6359 :*
6360 :*IN THIS TEST WE'LL MAKE SURE THAT 'BRANCH FIELD H' DOESN'T
6361 :*GET SUCH HIGH.
6362 :*FIRST WE'LL CLEAR THE PC HIGH REG. THEN WE'LL DO A BRANCH INSTR
6363 :*WITH BAB BITS 11&12 SET. IF PCR BITS 8&9 SET THEN WE'LL KNOW
6364 :*WE WERE SUCCESSFUL IF PCR BITS 8&9 FAIL TO SET, WE'LL KNOW
6365 :*THAT THE MUX SELECTED THE WRONG INPUT TO BE CLOCKED INTO THE PCR.
6366 :*
6367 030244 BADHEAD
6368 :***** TEST 39 *****
6369 :*
6370 030244 BGNTST
6371 030244 T39::
6372 030244 000240 NOP
6373 030246 000240 NOP
6374 030250 10$: MYINT ;INITIALIZE PARAMETERS
6375 030254 MSTCLR ;CLEAR DEVICE.
6376
6377 030260 SRMCLK ;DO A 'BRANCH ALWAYS' WITH
6378 030264 114400 114400 ;BAB BITS 11&12 SET THIS SHOULD CLOCK
6379 ;THESE BITS INTO BITS 8&9 OF THE PCR.
6380 030266 ROMCLK ;NOW READ THE PCR HIGH
6381 030272 121265 121265 ;AND PUT INTO PORT5.
6382 ;REG. BR NO CLK OF BAB BITS
6383 030274 116105 000005 MOVB 5(R1),R5 ;READ THE PCR.
6384 030300 112704 000003 MOVB #3,R4 ;EXPECT BITS 8,9 TO BE SET.
6385 030304 042705 000374 BIC #374,R5 ;STRIP ANY JUNK
6386 030310 020405 CMP R4,R5 ;OK?
6387 030312 001406 BEQ 20$
6388
6389 030314 ERROR 15 ;'BRANCH FIELD H' STUCK HIGH OR
6390 030320 104455 TRAP C$ERDF
6391 030322 000017 .WORD 15
6392 030324 005425 .WORD EMO
6393 030326 010710 .WORD ERR15
6394 ;OTHER PROBLEM IN THIS AREA.
6395
6396 030330 20$:
6397 030330 CHKREG
6398 .LIST ME
6399 030330 004737 003014 JSR PC,RDKM ; READ KMC11 REGISTERS
6400 030334 012702 030346 MOV #64$,R2 ; LOAD POINTER
6401 030340 004737 003120 JSR PC,R$CHECK ; CHECK FOR NO CHANGE
6402 030344 000405 BR 65$ ; SKIP LIST
6403 030346 000 000 000 64$: .BYTE .....20
6404 030351 000 000 000
6405 030354 000 000 000
6406 030357 020
6407 030360 65$:
6408 030360 ENDTST
6409 030360 L10110:
6410 030360 104401 TRAP C$ETST
```



```

6411 030362          BADHEAD
6412                :***** TEST 40 *****
6413                :*
6414                :*IN THIS TEST WE'RE GOING TO MAKE SURE THAT ONLY SPO
6415                :*IS SELECTED FOR SOURCE WHEN THE DESTINATION
6416                :*IS THE OUTBUS
6417                :*FIRST WE'LL WRITE EACH SP ADDR INTO ITSELF THEN WE'LL
6418                :*MOV SP TO OBUS4. THAT SHOULD SELECT
6419                :*SP ADDRESS 0. IF ANY OTHER DATA SHOWS UP, WE'LL
6420                :*BLAME IT ON THE SELECTION OF A DIFFERENT SCRATCH PAD.
6421 030362          BADHEAD
6422                :***** TEST 40 *****
6423
6424 030362          BGNTST
6425 030362          T40::
6426 030362          MYINT
6427 030366          MSTCLR
6428 030372 005005   CLR      R5          ;START WITH ADDR-ZERO
6429 030374 000137 030470 JMP      30$         ;***; SKIP FOR NOW
6430
6431 030400 042737 000017 030422 10$: BIC      #17,20$      ;STRIP SP ADDR FIELD FROM INSTR
6432 030406 010561 000004          MOV      R5,4(R1)    ;PUT SP ADDR INTO PORT4.
6433 030412 050537 030422          BIS      R5,20$      ;ADD SP ADDR TO INSTR.
6434 030416
6435 030422 123100          20$: ROMCLK   123100      ;WRITE TO SP
6436 030424 005205          INC      R5          ;UPDATE ADDRESS
6437 030426 120527 000020          CMPB   R5,#20       ;IF NOT THROUGH, REPFAT.
6438 030432 001362          BNE     10$
6439
6440 030434          ROMCLK          ;NOW MOV SPO TO OBUS* PORT4
6441 030440 061204          061204      ;
6442 030442 016104 000004          MOV      4(R1),R4    ;READ PORT4 IT S/B ZERO
6443 030446 001410          BEQ     30$
6444 030450 012705 000000          MOV      #0,R5
6445 030454          ERROR   43          ;SPO NOT SELECTED FOR SOURCE-SEE
6446 030460 104455          TRAP   C$ERDF
6447 030462 000053          .WORD  43
6448 030464 005425          .WORD  EMO
6449 030466 012130          .WORD  ERR43
6450
6451                ;DISCUSSION IN HEADER.
6452 030470          30$:
6453 030470
6454
6455 030470 004737 003014          CHKREG  .LIST   ME
6456 030474 012702 030506          JSR    PC,RDKM    ; READ KMC11 REGISTERS
6457 030500 004737 003120          MOV    #64$,R2   ; LOAD POINTER
6458 030504 000405          JSR    PC,RCHECK  ; CHECK FOR NO CHANGE
6459 030506          000      000      000 64$: .BYTE   .....20 ; SKIP LIST
6460 030511          000      000      000
6461 030514          000      000      000
6462 030517          020
6463 030520          65$:
6464 030520          L10111: ENDTST
6465 030520
6466 030520 104401          TRAP   C$ETST

```

```

6467 030522          BADHEAD
6468                :***** TEST 41 *****
6469                :*
6470                :*IN THIS TEST WE ARE GOING TO MAKE SURE THAT THE
6471                :*SIGNAL 'MOV INST H' (AND ITS ASSOC. TRIBS) DOESN'T GET
6472                :*STUCK HIGH. IN ORDER TO DO THIS WE'LL CLEAR THE PC HIGH REG
6473                :*PUT KNOWN DATA IN THE BREG AND SP1 THEN WE'LL A BRANCH
6474                :*WITH CROM BITS 0-3 SET AS WELL AS CROM BIT 9 WITH CROM BITS 8 AND 11 CLEAR.
6475                :*IF 'MOV INST H' GETS STUCK HIGH, THE PC REG HIGH WILL GET LOADED
6476                :*WITH THE CONTENTS OF THE ALU
6477 030522          BADHEAD
6478                :***** TEST 41 *****
6479
6480 030522          BGNTST
6481 030522          T41::
6482 030522          000240      NOP
6483 030524          000240      NOP
6484 030526          10$:      MYINT          ;DO INITIAL TEST SET-UP.
6485 030532          MSTCLR          ;DO A MASTER CLEAR.
6486 030536          012761 000002 000004      MOV          #2,4(R1)      ;PUT A 2 INTO SP1
6487 030544          ROMCLK          ;PORT4 TO SCRATCH PAD 1
6488 030550          123101          123101
6489 030552          012761 000004 000004      MOV          #4,4(R1)
6490 030560          ROMCLK          123101
6491 030564          123100          123100
6492 030566          SROMCLK
6493 030572          141201          141201      ;NOW DO A BRANCH ON C-BIT SET
6494                ;BASED ON SP CONTENTS
6495                ;OK-WHAT WE ARE REALLY
6496                ;INTERESTED IN IS SEEING IF THE
6497                ;PC HIGH REG GETS LOADED WITH
6498                ;THE CONTENTS OF THE ALU (2)
6499                ;IF THIS OCCURS, WE CAN PROBABLY
6500                ;SAY THAT 'MOV INSTR' REMAINED
6501 030574          ROMCLK          ;HIGH.
6502 030600          121265          ;READ PC HIGH, PUT INTO PORTS
6503 030602          116104 000005      MOVB          5(R1),R4      ;READ PC REG HIGH FROM PORT
6504 030606          001407          BEQ          20$          ;SHOULD BE CLEAR
6505 030610          005005          CLR          R5
6506
6507 030612          ERROR          15      ;ERROR-PC REG HIGH S/B CLEAR-SEE HEADER
6508 030616          104455          TRAP          C$ERDF
6509 030620          000017          .WORD          15
6510 030622          005425          .WORD          EMO
6511 030624          010710          .WORD          ERR15
6512
6513                ;DISCUSSION.
6514 030626          20$:
6515 030626          CHKREG
6516                .LIST          ME
6517 030626          004737 003014      JSR          PC,RDKM      ; READ KMC11 REGISTERS
6518 030632          012702 030644      MOV          #64$,R2      ; LOAD POINTER
6519 030636          004737 003120      JSR          PC,R$CHECK   ; CHECK FOR NO CHANGE
6520 030642          000405          BR          65$          ; SKIP LIST
6521 030644          000          000          000 64$:      .BYTE          .....20
6522 030647          000          000          000

```

6523	030652	000	000	000
6524	030655	020		
6525	030656			
6526	030656			
6527	030656			
6528	030656	104401		

65\$: ENDTST
L10112: TRAP C\$ETST


```
6529 030660 BADHEAD
6530 :***** TEST 42 *****
6531 :*TEST THAT MASTER CLEAR, CLEARS BITS IN THE NPR CONTROL REGISTER AND
6532 :*MICROPROCESSOR MISCELLANEOUS REGISTER-FIRST WE'LL SET THE
6533 :*PRIORITY UP SO THAT WHEN WE SET THE BUS REQUEST BIT THAT IT WON'T BUG US
6534 :*THEN WE'LL SET ALL THE BITS IN BOTH REGS EXCEPT THE
6535 :*NPR REQUEST. WE'LL LOOK TO SEE THAT ALL GOT SET, NEXT
6536 :*WE'LL DO A MASTER CLEAR AND BE SURE THAT THEY ALL
6537 :*CLEAR.
6538 030660 BADHEAD
6539 :***** TEST 42 *****
6540
6541 030660 BGNTST
6542 030660 T42::
6543 030660 MYINT
6544 030664 MSTCLR
6545 030670 000240 NOP
6546 030672 000240 NOP
6547 030674 SETPRI #PRI07 ;DON'T ALLOW INTERRUPTS.
6548 030674 012700 000340 MOV #PRI07,R0
6549 030700 104441 TRAP C$SPRI
6550 030702 012761 000354 000004 MOV #354,4(R1) ;DATA TO BE SET
6551 030710 ROMCLK
6552 030714 121111 121111 ;PUT INTO MISC REG.
6553 030716 012761 000236 000004 MOV #236,4(R1) ; DO NOT SET NPR REQ OR TIM
6554 030724 ROMCLK
6555 030730 121110 121110 ;PUT INTO NPR REG
6556 030732 ROMCLK
6557 030736 121225 121225 ;MOV MISC REG (11) TO PORT5
6558
6559 030740 ROMCLK
6560 030744 121204 121204 ;MOVE NPR REG (10) TO PORT4
6561 030746 012737 166236 002414 MOV #166236,$GDDAT ;EXPECT ALL TO SET
6562 030754 016104 000004 MOV 4(R1),R4 ;READ WHAT HAPPEN
6563 030760 042704 010000 BIC #10000,R4 ; CLEAR TIMER IF SET
6564 030764 023704 002414 CMP $GDDAT,R4 ;DID ALL BITS GET SET?
6565 030770 001410 BEQ 10$ ;YES CONTINUE.
6566 030772 BRESET
6567 030772 104433 TRAP C$RESET
6568 030774 ERROR 46 ;SOME SORT OF PROBLEM SETTING BITS
6569 031000 104455 TRAP C$ERDF
6570 031002 000056 .WORD 46
6571 031004 005425 .WORD EMO
6572 031006 012316 .WORD ERR46
6573 ;IN THE NPR AND/OR MISC REG.
6574 031010 CKLOOP
6575 031010 104406 TRAP C$CLP1
6576
6577 031012 005011 10$: CLR (R1)
6578 031014 152761 000100 000001 BISB #100,1(R1) ;SET MASTER CLEAR
6579 031022 142761 000100 000001 BICB #100,1(R1) ;CLEAR MASTER CLEAR
6580
6581 031030 ROMCLK
6582 031034 121225 121225 ;MOV MISC REG (11) TO PORT5
6583
6584 031036 ROMCLK
```


6641 031222
6642 031222
6643
6644 031222 004737 003014
6645 031226 012702 031240
6646 031232 004737 003120
6647 031236 000405
6648 031240 000 000
6649 031243 000 000
6650 031246 000 000
6651 031251 020
6652 031252
6653 031252
6654 031252
6655 031252 104401

40\$:

CHKREG

.LIST

JSR

MOV

JSR

BR

.BYTE

ME

PC,RDKM

#64\$,R2

PC,RCHECK

65\$

.....20

: READ KMC11 REGISTERS

: LOAD POINTER

: CHECK FOR NO CHANGE

: SKIP LIST

64\$:

65\$:

L10113:

ENDTST

TRAP

CSETST

.SBTTL HARDWARE PARAMETER CODING SECTION

6656
6657
6658
6659
6660
6661
6662
6663
6664
6665
6666
6667
6668
6669 031254
6670 031254 000011
6671 031256
6672
6673
6674 031256
6675 031256 001031
6676 031260 031362
6677 031262 160000
6678 031264 177776
6679
6680
6681
6682
6683
6684
6685 031266
6686 031266 011032
6687 031270 031721
6688 031272 000001
6689 031274 000000
6690 031276 000001
6691
6692 031300
6693
6694 031300
6695
6696 031300 044127 041511 020110
6697 031306 044515 051103 026517
6698 031314 050103 037525 024040
6699 031322 036460 034115 030062
6700 031330 026060 036464 034115
6701 031336 030062 026064 036466
6702 031344 034115 030062 026066
6703 031352 036467 034115 030062
6704 031360 000067
6705 031362 051503 020122 042101
6706 031370 051104 051505 020123
6707 031376 020072 000
6708 031401 126 041505 047524
6709 031406 020122 042101 051104
6710 031414 051505 020123 020072
6711 031422 000

:/ 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.

BGNHRD
.WORD L10114-L\$HARD/2
L\$HARD::
:
GPRMD WPM,0,0,7,0,7,YES
GPRMA ADDRES,2,0,160000,177776,YES
.WORD T\$CODE
.WORD ADDRES
.WORD T\$LOLIM
.WORD T\$HILIM
:
GPRMA VECTOR,4,0,0,674,YES
GPRMD PRIRTY,6,0,7000,4,7,YES
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,20,0,40000,0,1,YES
GPRMD PFTST,22,0,1,0,1,YES
.WORD T\$CODE
.WORD PFTST
.WORD 1
.WORD T\$LOLIM
.WORD T\$HILIM
ENDHRD
.EVEN
L10114:
WPM: .ASCIZ 'WHICH MICRO-CPU? (0=M8200,4=M8204,6=M8206,7=M8207)'
ADDRES: .ASCIZ /CSR ADDRESS : /
VECTOR: .ASCIZ /VECTOR ADDRESS : /

6712 031423 120 044522 051117 PRIRTY: .ASCIZ /PRIORITY LEVEL : /
6713 031430 052111 020131 042514
6714 031436 042526 020114 020072
6715 031444 000
6716 031445 127 044510 044103 LNUNIT: .ASCIZ /WHICH LINE UNIT (0-3)? 0=NONE,1=M8201,2=M8202,3=M8203 : /
6717 031452 046040 047111 020105
6718 031460 047125 052111 024040
6719 031466 026460 024463 020077
6720 031474 036460 047516 042516
6721 031502 030454 046475 031070
6722 031510 030460 031054 046475
6723 031516 031070 031060 031454
6724 031524 046475 031070 031460
6725 031532 035040 000040
6726 031536 053523 052111 044103 SWPAC1: .ASCIZ /SWITCH PACK #1 (DDCMP LINE #) : /
6727 031544 050040 041501 020113
6728 031552 030443 024040 042104
6729 031560 046503 020120 044514
6730 031566 042516 021440 020051
6731 031574 020072 000
6732 031577 123 044527 041524 SWPAC2: .ASCIZ /SWITCH PACK #2 (BM873 BOOT ADR) : /
6733 031604 020110 040520 045503
6734 031612 021440 020062 041050
6735 031620 034115 031467 041040
6736 031626 047517 020124 042101
6737 031634 024522 035040 000040
6738 031642 044527 046114 052040 LOOPBK: .ASCIZ /WILL TEST CONNECTOR(S) BE USED ? 0=NO,1=YES : /
6739 031650 051505 020124 047503
6740 031656 047116 041505 047524
6741 031664 024122 024523 041040
6742 031672 020105 051525 042105
6743 031700 037440 030040 047075
6744 031706 026117 036461 042531
6745 031714 020123 020072 000
6746 031721 122 047125 051040 PFTST: .ASCIZ /RUN REMOTE POWERFAIL TEST ? 0=NO, 1=YES :/
6747 031726 046505 052117 020105
6748 031734 047520 042527 043122
6749 031742 044501 020114 042524
6750 031750 052123 037440 030040
6751 031756 047075 026117 030440
6752 031764 054475 051505 035040
6753 031772 000
6754
6755 031774 .EVEN
6756
6757
6758
6759
6760
6761

.SBTTL SOFTWARE PARAMETER CODING SECTION

6762
6763
6764
6765
6766
6767
6768
6769
6770
6771
6772
6773
6774
6775
6776
6777
6778
6779
6780
6781
6782
6783
6784
6785
6786
6787
6788
6789
6790
6791
6792
6793
6794
6795
6796

031774
031774 000000
031776

031776
031776

031776
031776 000000
032000 000000
032002

000001

```
:/ 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 L10115-L$SOFT/2  
L$SOFT::  
  
ENDSFT  
.EVEN  
L10115:  
.EVEN  
  
LASTAD  
.EVEN  
.WORD 0  
.WORD 0  
L$LAST::  
.END
```


Symbol	682#	2974	3029	3086	3110	3590	3614	3638	3693	3714	3735	3789	3812
C\$BRK = 000022	682#												
C\$BSEG= 000004	682#	2974	3029	3086	3110	3590	3614	3638	3693	3714	3735	3789	3812
	3835	3890	3912	3934	3990	4012	4034	4090	4112	4134	4189	4211	4233
	4288	4310	4332	4386	4408	4432	4486	4508	4532	4586	4608	4632	4686
	4708	4732	4786	4808	4832	4886	4908	4932					
C\$BSUB= 000002	682#												
C\$CEFG= 000045	682#												
C\$CLCK= 000062	682#												
C\$CLEA= 000012	682#	2807											
C\$CLOS= 000035	682#												
C\$CLP1= 000006	682#	6575	6599	6620									
C\$CVEC= 000036	682#												
C\$DCLN= 000044	682#												
C\$DODU= 000051	682#												
C\$DRPT= 000024	682#												
C\$DU = 000053	682#	2827											
C\$EDIT= 000003	682#	759											
C\$ERDF= 000055	682#	2878	2920	2938	2983	3039	3099	3122	3185	3249	3313	3342	3404
	3426	3466	3514	3535	3551	3603	3627	3649	3703	3724	3745	3800	3823
	3846	3901	3923	3945	4001	4023	4045	4101	4123	4145	4200	4222	4244
	4299	4321	4343	4397	4421	4443	4497	4521	4543	4597	4621	4643	4697
	4721	4743	4797	4821	4843	4897	4921	4943	5016	5084	5184	5257	5300
	5318	5334	5379	5396	5412	5455	5471	5485	5538	5560	5571	5588	5601
	5721	5747	5766	5785	5869	5928	6023	6038	6053	6067	6083	6095	6107
	6119	6223	6253	6267	6327	6390	6446	6508	6569	6593	6613	6633	
C\$ERHR= 000056	682#												
C\$ERRO= 000060	682#												
C\$ERSF= 000054	682#												
C\$ERSO= 000057	682#												
C\$ESCA= 000010	682#	2869	2886	2926	2988	3044	3104	3127	3190	3254	3318	3347	3434
	3519	3608	3632	3654	3708	3729	3750	3806	3829	3851	3906	3928	3950
	4006	4028	4050	4106	4128	4150	4205	4227	4249	4304	4326	4348	4402
	4426	4448	4502	4526	4548	4602	4626	4648	4702	4726	4748	4802	4826
	4848	4902	4926	4948	5025	5092	5219	5306	5324	5385	5402	5935	6029
	6045	6059	6073	6089	6100	6112	6258	6334					
C\$ESEG= 000005	682#	2992	3048	3108	3131	3612	3636	3658	3712	3733	3754	3810	3833
	3855	3910	3932	3954	4010	4032	4054	4110	4132	4154	4209	4231	4253
	4308	4330	4352	4406	4430	4452	4506	4530	4552	4606	4630	4652	4706
	4730	4752	4806	4830	4852	4906	4930	4952					
C\$ESUB= 000003	682#												
C\$ETST= 000001	682#	2891	2957	3013	3070	3150	3213	3277	3366	3484	3569	3672	3768
	3869	3968	4068	4168	4267	4366	4466	4566	4666	4766	4866	4966	5044
	5113	5276	5352	5430	5504	5613	5828	5888	5961	6159	6231	6287	6356
	6410	6466	6528	6655									
C\$EXIT= 000032	682#	5366	5579	5804	5952	6137	6204						
C\$GETB= 000026	682#												
C\$GETW= 000027	682#												
C\$GMAN= 000043	682#												
C\$GPHR= 000042	682#	2699											
C\$GPLO= 000030	682#												
C\$GPRI= 000040	682#												
C\$INIT= 000011	682#	2786											
C\$INLP= 000020	682#												
C\$MANI= 000050	682#												
C\$MEM = 000031	682#												
C\$MSG = 000023	682#	1894	1930	1966	2001	2036	2072	2108	2143	2178	2213	2248	2283

EM29	006036	1770#	2394																	
EM3	005501	1727#	1976																	
EM35	005760	1760#	2429																	
EM4	005514	1729#	2011																	
EM5	005541	1733#	2046																	
EM6	005566	1737#	2082																	
EM7	005603	1740#	2118	2188																
ENDBUG	003634	1442#																		
ENDIT	013104	2679	2774	2778	2781	2783#														
ERRFLG	002312	1077#																		
ERR1	006274 G	1860#	2986	3042	3102	5931														
ERR10	007666 G	2110#																		
ERR11	010040 G	2145#	3407	3429																
ERR12	010212 G	2180#	2923	2941	5087															
ERR13	010364 G	2215#	5019																	
ERR14	010536 G	2250#	5458	5474	5488															
ERR15	010710 G	2285#	5187	6330	6393	6511														
ERR16	011066 G	2321#	5303	5321	5337	5382	5399	5415												
ERR17	011240 G	2358#	5541	5563	5604															
ERR2	006452 G	1896#	3125																	
ERR29	011362 G	2386#	5724	5750	6026	6041	6056	6070	6086	6098	6110	6122								
ERR3	006630 G	1932#																		
ERR35	011534 G	2421#	5574	5591																
ERR36	011706 G	2457#	2881																	
ERR4	007006 G	1968#																		
ERR40	011760 G	2477#	6226																	
ERR41	012024 G	2495#	6256																	
ERR42	012066 G	2512#	6270																	
ERR43	012130 G	2530#	6449																	
ERR44	012176 G	2549#	3469																	
ERR45	012246 G	2568#	5260																	
ERR46	012316 G	2587#	6572	6596	6616	6636														
ERR5	007160 G	2003#	3606	3630	3652	3706	3727	3748	3803	3826	3849	3904	3926	3948						
		4004	4026	4048	4104	4126	4148	4203	4225	4247	4302	4324	4346	4'00						
		4424	4446	4500	4524	4546	4600	4624	4646	4700	4724	4746	4800	4824						
		4846	4900	4924	4946															
ERR6	007332 G	2038#	3188	3252	3316	3345	5788													
ERR7	007510 G	2074#	3517	3538	3554	5769	5872													
EVL =	000004 G	985#																		
E\$END =	002100	682#																		
E\$LOAD =	000035	682#	782																	
FADR	002354	1094#	2307	5156*	5159	5160	5225*	5226*	5227	6310*	6312	6337*	6338*							
FLAG	002350	1092#	2058	3163*	3167	3168	3196*	3197	3226*	3231	3232	3260*	3261	3291*						
		3292	3295	3296	3320*	3321	3324*	3325	3328	3349*	3350									
FMX	005346	1707#	1863	1899	1935	1971	2006	2041	2077	2113	2148	2183	2218	2253						
		2288	2324	2361	2389	2424														
FM1	005337	1705#	1869	1875	1905	1911	1941	1947	1977	1983	2012	2018	2047	2053						
		2083	2089	2119	2125	2154	2160	2189	2195	2224	2230	2259	2265	2294						
		2300	2330	2336	2367	2373	2395	2401	2430	2436										
FTIME	002424	1114#	2655	2659*																
F\$AU =	000015	682#	2842	2844																
F\$AUTO =	000020	682#	2635	2637																
F\$BGN =	000040	682#	688	1860	1896	1932	1968	2003	2038	2074	2110	2145	2180	2215						
		2250	2285	2321	2358	2386	2421	2457	2477	2495	2512	2530	2549	2568						
		2587	2615	2621	2628	2635	2649	2802	2821	2842	2861	2869	2886	2890						
		2902	2926	2956	2966	2974	2988	3012	3022	3029	3044	3069	3080	3086						

3104	3110	3127	3149	3159	3190	3212	3222	3254	3276	3287	3318	3347
3365	3376	3434	3483	3495	3519	3568	3584	3590	3608	3614	3632	3638
3654	3671	3687	3693	3708	3714	3729	3735	3750	3767	3783	3789	3806
3812	3829	3835	3851	3868	3884	3890	3906	3912	3928	3934	3950	3967
3983	3990	4006	4012	4028	4034	4050	4067	4083	4090	4106	4112	4128
4134	4150	4167	4183	4189	4205	4211	4227	4233	4249	4266	4282	4288
4304	4310	4326	4332	4348	4365	4381	4386	4402	4408	4426	4432	4448
4465	4481	4486	4502	4508	4526	4532	4548	4565	4581	4586	4602	4608
4626	4632	4648	4665	4681	4686	4702	4708	4726	4732	4748	4765	4781
4786	4802	4808	4826	4832	4848	4865	4881	4886	4902	4908	4926	4932
4948	4965	4979	5025	5043	5058	5092	5112	5151	5219	5275	5288	5306
5324	5351	5364	5366	5385	5402	5429	5441	5503	5519	5579	5612	5626
5804	5827	5840	5887	5899	5935	5952	5960	5972	6029	6045	6059	6073
6089	6100	6112	6137	6158	6172	6204	6230	6243	6258	6286	6299	6334
6355	6371	6409	6425	6465	6481	6527	6542	6654	6670	6775		
682#	2802	2806										
682#	2821	2826										
682#	688	1895	1931	1967	2002	2037	2073	2109	2144	2179	2214	2249
2284	2320	2355	2385	2420	2455	2475	2494	2511	2528	2548	2567	2586
2605	2619	2626	2639	2787	2808	2828	2846	2861	2869	2886	2890	2892
2902	2926	2956	2958	2966	2988	2993	3012	3014	3022	3044	3049	3069
3071	3080	3104	3109	3127	3132	3149	3151	3159	3190	3212	3214	3222
3254	3276	3278	3287	3318	3347	3365	3367	3376	3434	3483	3485	3495
3519	3568	3570	3584	3608	3613	3632	3637	3654	3659	3671	3673	3687
3708	3713	3729	3734	3750	3755	3767	3769	3783	3806	3811	3829	3834
3851	3856	3868	3870	3884	3906	3911	3928	3933	3950	3955	3967	3969
3983	4006	4011	4028	4033	4050	4055	4067	4069	4083	4106	4111	4128
4133	4150	4155	4167	4169	4183	4205	4210	4227	4232	4249	4254	4266
4268	4282	4304	4309	4326	4331	4348	4353	4365	4367	4381	4402	4407
4426	4431	4448	4453	4465	4467	4481	4502	4507	4526	4531	4548	4553
4565	4567	4581	4602	4607	4626	4631	4648	4653	4665	4667	4681	4702
4707	4726	4731	4748	4753	4765	4767	4781	4802	4807	4826	4831	4848
4853	4865	4867	4881	4902	4907	4926	4931	4948	4953	4965	4967	4979
5025	5043	5045	5058	5092	5112	5114	5151	5219	5275	5277	5288	5306
5324	5351	5353	5364	5366	5385	5402	5429	5431	5441	5503	5505	5519
5579	5612	5614	5626	5804	5827	5829	5840	5887	5889	5899	5935	5952
5960	5962	5972	6029	6045	6059	6073	6089	6100	6112	6137	6158	6160
6172	6204	6230	6232	6243	6258	6286	6288	6299	6334	6355	6357	6371
6409	6411	6425	6465	6467	6481	6527	6529	6542	6654	6656	6695	6782
682#	6670	6693										
682#	874	889										
682#	2649	2785										
682#	2619	5366	5579	5804	5952	6137	6204					
682#	688											
682#	1860	1893	1896	1929	1932	1965	1968	2000	2003	2035	2038	2071
2074	2107	2110	2142	2145	2177	2180	2212	2215	2247	2250	2282	2285
2318	2321	2353	2358	2383	2386	2418	2421	2453	2457	2473	2477	2492
2495	2509	2512	2526	2530	2546	2549	2565	2568	2584	2587	2603	
682#	2628	2633										
682#												
682#	2615	2624										
682#	2974	2991	3029	3047	3086	3107	3110	3130	3590	3611	3614	3635
3638	3657	3693	3711	3714	3732	3735	3753	3789	3809	3812	3832	3835
3854	3890	3909	3912	3931	3934	3953	3990	4009	4012	4031	4034	4053
4090	4109	4112	4131	4134	4153	4189	4208	4211	4230	4233	4252	4288
4307	4310	4329	4332	4351	4386	4405	4408	4429	4432	4451	4486	4505

FSCLEA= 000007
FSDU = 000016
FSEND = 000041

FSHARD= 000004
FSHW = 000013
FSINIT= 000006
FSJMP = 000050
FSMOD = 000000
FSMSG = 000011

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

CZKMCAO KMC11-B STATIC PART2
CZKMCA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 PAGE 175
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0174

LSSOFT	031776	G	6775	6776#		
LSSPC	002056	G	763#			
LSSPCP	002020	G	733#			
LSSPTP	002024	G	737#			
LSSTA	002030	G	741#			
LSSW	002300	G	738	898	899#	
LSTEST	002114	G	793#			
LSTIML	002014	G	729#			
LSUNIT	002012	G	727#			
L10000	002276		874	889#		
L10001	002300		898	904#		
L10002	006450		1893#			
L10003	006626		1929#			
L10004	007004		1965#			
L10005	007156		2000#			
L10006	007330		2035#			
L10007	007506		2071#			
L10010	007664		2107#			
L10011	010036		2142#			
L10012	010210		2177#			
L10013	010362		2212#			
L10014	010534		2247#			
L10015	010706		2282#			
L10016	011064		2318#			
L10017	011236		2353#			
L10020	011360		2383#			
L10021	011532		2418#			
L10022	011704		2453#			
L10023	011756		2473#			
L10024	012022		2492#			
L10025	012064		2509#			
L10026	012126		2526#			
L10027	012174		2546#			
L10030	012244		2565#			
L10031	012314		2584#			
L10032	012364		2603#			
L10033	012372		2620	2624#		
L10035	012402		2637#			
L10036	013104		2785#			
L10037	013106		2806#			
L10040	013112		2826#			
L10041	013114		2844#			
L10042	013224		2870	2887	2890#	
L10043	013416		2927	2956#		
L10044	013570		3012#			
L10045	013746		3069#			
L10046	014226		3149#			
L10047	014456		3191	3212#		
L10050	014716		3255	3276#		
L10051	015266		3319	3348	3365#	
L10052	015646		3435	3483#		
L10053	016132		3520	3568#		
L10054	016416		3671#			
L10055	016662		3767#			
L10056	017142		3868#			
L10057	017422		3967#			

SIB3	003163	1376#												
SIB4	003164	1350*	1377#											
SIB5	003165	1378#												
SIB6	003166	1351*	1379#											
SIB7	003167	1380#												
SINGLE	002734	1317#	1331	1342	1348	1354								
SKIP33	002432	1117#	2756*	2759*	5523									
SPBR =	003400 G	1022#												
SPR =	003000 G	1021#												
SRG0 =	000000 G	1035#	1402											
SRG1 =	000020 G	1036#	1403											
SRG10 =	000200 G	1043#	1414											
SRG11 =	000220 G	1044#	1415											
SRG12 =	000240 G	1045#												
SRG13 =	000260 G	1046#												
SRG14 =	000300 G	1047#												
SRG15 =	000320 G	1048#												
SRG16 =	000340 G	1049#												
SRG17 =	000360 G	1050#												
SRG2 =	000040 G	1037#	1404											
SRG3 =	000060 G	1038#	1405											
SRG4 =	000100 G	1039#	1408											
SRG5 =	000120 G	1040#	1409											
SRG6 =	000140 G	1041#	1410											
SRG7 =	000160 G	1042#	1411											
SSTACK	002670	1170#	2652											
STAT	002320	1080#												
STAT1	002440	1147#	2731*	2733*	2737*	2748*	2752*	2755*						
STAT2	002442	1148#	2762*	2764*										
STAT3	002444	1149#												
STM	006071	1775#	1888	1924	1960	1995	2030	2066	2102	2137	2172	2207	2242	2277
		2313	2348	2378	2413	2448	2467	2486	2503	2520	2540	2559	2578	2597
STRTSW	002316	1079#												
SUBRPC	002310	1076#												
SVCGBL=	000000	682#	688	695#	714	715	723	724	725	726	727	728	729	730
		731	732	733	734	735	736	737	738	739	740	741	742	743
		744	745	746	747	748	749	750	751	752	753	754	755	756
		757	758	760	761	763	764	765	766	767	768	769	770	771
		772	773	774	775	776	777	778	779	780	781	782	783	784
		785	786	787	788	789	790	791	792	793	794	795	796	797
		798	815	816	875	876	877	899	900	901	1190	1191	1194	1195
		1860	1861	1896	1897	1932	1933	1968	1969	2003	2004	2038	2039	2074
		2075	2110	2111	2145	2146	2180	2181	2215	2216	2250	2251	2285	2286
		2321	2322	2358	2359	2386	2387	2421	2422	2457	2458	2477	2478	2495
		2496	2512	2513	2530	2531	2549	2550	2568	2569	2587	2588	2615	2616
		2628	2629	2635	2636	2649	2650	2802	2803	2821	2822	2842	2843	6671
		6672	6776	6777	6794#	6795								
SVCINS=	000000	682#	692#	715	716	717	718	719	720	721	722	723	724	725
		726	727	728	729	730	731	732	733	734	735	736	737	738
		739	740	741	742	743	744	745	746	747	748	749	750	751
		752	753	754	755	756	757	758	759	760	761	762	763	764
		765	766	767	768	769	770	771	772	773	774	775	776	777
		778	779	780	781	782	783	784	785	786	787	788	789	790
		791	792	793	794	795	796	797	798	799	814	815	816	817
		818	819	820	821	822	823	824	825	826	827	828	829	830
		831	832	833	834	835	836	837	838	839	840	841	842	843

844	845	846	847	848	849	850	851	852	853	854	855	856
857	858	874	875	898	899	1191	1192	1193	1195	1200	1201	1861
1862	1863	1864	1865	1866	1867	1868	1869	1870	1871	1872	1873	1874
1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887
1888	1889	1890	1891	1892	1893	1894	1895	1897	1898	1899	1900	1901
1902	1903	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914
1915	1916	1917	1918	1919	1920	1921	1922	1923	1924	1925	1926	1927
1928	1929	1930	1931	1933	1934	1935	1936	1937	1938	1939	1940	1941
1942	1943	1944	1945	1946	1947	1948	1949	1950	1951	1952	1953	1954
1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967
1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981
1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994
1995	1996	1997	1998	1999	2000	2001	2002	2004	2005	2006	2007	2008
2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034
2035	2036	2037	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048
2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061
2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2075
2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088
2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	2101
2102	2103	2104	2105	2106	2107	2108	2109	2111	2112	2113	2114	2115
2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128
2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141
2142	2143	2144	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155
2156	2157	2158	2159	2160	2161	2162	2163	2164	2165	2166	2167	2168
2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2181	2182
2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193	2194	2195
2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208
2209	2210	2211	2212	2213	2214	2216	2217	2218	2219	2220	2221	2222
2223	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235
2236	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248
2249	2251	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	2262
2263	2264	2265	2266	2267	2268	2269	2270	2271	2272	2273	2274	2275
2276	2277	2278	2279	2280	2281	2282	2283	2284	2286	2287	2288	2289
2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302
2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315
2316	2317	2318	2319	2320	2322	2323	2324	2325	2326	2327	2328	2329
2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342
2343	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355
2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371
2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384
2385	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398
2399	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411
2412	2413	2414	2415	2416	2417	2418	2419	2420	2422	2423	2424	2425
2426	2427	2428	2429	2430	2431	2432	2433	2434	2435	2436	2437	2438
2439	2440	2441	2442	2443	2444	2445	2446	2447	2448	2449	2450	2451
2452	2453	2454	2455	2459	2460	2461	2462	2463	2464	2465	2466	2467
2468	2469	2470	2471	2472	2474	2475	2479	2480	2481	2482	2483	2484
2485	2486	2487	2488	2489	2490	2491	2493	2494	2497	2498	2499	2500
2501	2502	2503	2504	2505	2506	2507	2508	2510	2511	2514	2515	2516
2517	2518	2519	2520	2521	2522	2523	2524	2525	2527	2528	2532	2533
2534	2535	2536	2537	2538	2539	2540	2541	2542	2543	2544	2545	2547
2548	2551	2552	2553	2554	2555	2556	2557	2558	2559	2560	2561	2562
2563	2564	2566	2567	2570	2571	2572	2573	2574	2575	2576	2577	2578
2579	2580	2581	2582	2583	2585	2586	2589	2590	2591	2592	2593	2594
2595	2596	2597	2598	2599	2600	2601	2602	2604	2605	2619	2620	2621

2625	2626	2638	2639	2665	2666	2667	2668	2669	2671	2672	2673	2674
2675	2676	2677	2678	2679	2680	2683	2684	2685	2686	2687	2698	2699
2700	2701	2702	2703	2786	2787	2807	2808	2824	2825	2827	2828	2845
2846	2869	2870	2871	2878	2879	2880	2881	2882	2886	2887	2888	2891
2892	2920	2921	2922	2923	2924	2926	2927	2928	2938	2939	2940	2941
2942	2957	2958	2974	2975	2983	2984	2985	2986	2987	2988	2989	2990
2992	2993	3013	3014	3029	3030	3039	3040	3041	3042	3043	3044	3045
3046	3048	3049	3070	3071	3086	3087	3099	3100	3101	3102	3103	3104
3105	3106	3108	3109	3110	3111	3122	3123	3124	3125	3126	3127	3128
3129	3131	3132	3150	3151	3185	3186	3187	3188	3189	3190	3191	3192
3213	3214	3249	3250	3251	3252	3253	3254	3255	3256	3277	3278	3313
3314	3315	3316	3317	3318	3319	3320	3342	3343	3344	3345	3346	3347
3348	3349	3366	3367	3404	3405	3406	3407	3408	3426	3427	3428	3429
3430	3434	3435	3436	3466	3467	3468	3469	3470	3484	3485	3514	3515
3516	3517	3518	3519	3520	3521	3535	3536	3537	3538	3539	3551	3552
3553	3554	3555	3569	3570	3590	3591	3603	3604	3605	3606	3607	3608
3609	3610	3612	3613	3614	3615	3627	3628	3629	3630	3631	3632	3633
3634	3636	3637	3638	3639	3649	3650	3651	3652	3653	3654	3655	3656
3658	3659	3672	3673	3693	3694	3703	3704	3705	3706	3707	3708	3709
3710	3712	3713	3714	3715	3724	3725	3726	3727	3728	3729	3730	3731
3733	3734	3735	3736	3745	3746	3747	3748	3749	3750	3751	3752	3754
3755	3768	3769	3789	3790	3800	3801	3802	3803	3804	3806	3807	3808
3810	3811	3812	3813	3823	3824	3825	3826	3827	3829	3830	3831	3833
3834	3835	3836	3846	3847	3848	3849	3850	3851	3852	3853	3855	3856
3869	3870	3890	3891	3901	3902	3903	3904	3905	3906	3907	3908	3910
3911	3912	3913	3923	3924	3925	3926	3927	3928	3929	3930	3932	3933
3934	3935	3945	3946	3947	3948	3949	3950	3951	3952	3954	3955	3968
3969	3990	3991	4001	4002	4003	4004	4005	4006	4007	4008	4010	4011
4012	4013	4023	4024	4025	4026	4027	4028	4029	4030	4032	4033	4034
4035	4045	4046	4047	4048	4049	4050	4051	4052	4054	4055	4068	4069
4090	4091	4101	4102	4103	4104	4105	4106	4107	4108	4110	4111	4112
4113	4123	4124	4125	4126	4127	4128	4129	4130	4132	4133	4134	4135
4145	4146	4147	4148	4149	4150	4151	4152	4154	4155	4168	4169	4189
4190	4200	4201	4202	4203	4204	4205	4206	4207	4209	4210	4211	4212
4222	4223	4224	4225	4226	4227	4228	4229	4231	4232	4233	4234	4244
4245	4246	4247	4248	4249	4250	4251	4253	4254	4267	4268	4288	4289
4299	4300	4301	4302	4303	4304	4305	4306	4308	4309	4310	4311	4321
4322	4323	4324	4325	4326	4327	4328	4330	4331	4332	4333	4343	4344
4345	4346	4347	4348	4349	4350	4352	4353	4366	4367	4386	4387	4397
4398	4399	4400	4401	4402	4403	4404	4406	4407	4408	4409	4421	4422
4423	4424	4425	4426	4427	4428	4430	4431	4432	4433	4443	4444	4445
4446	4447	4448	4449	4450	4452	4453	4466	4467	4486	4487	4497	4498
4499	4500	4501	4502	4503	4504	4506	4507	4508	4509	4521	4522	4523
4524	4525	4526	4527	4528	4530	4531	4532	4533	4543	4544	4545	4546
4547	4548	4549	4550	4552	4553	4566	4567	4586	4587	4597	4598	4599
4600	4601	4602	4603	4604	4606	4607	4608	4609	4621	4622	4623	4624
4625	4626	4627	4628	4630	4631	4632	4633	4643	4644	4645	4646	4647
4648	4649	4650	4652	4653	4666	4667	4686	4687	4697	4698	4699	4700
4701	4702	4703	4704	4706	4707	4708	4709	4721	4722	4723	4724	4725
4726	4727	4728	4730	4731	4732	4733	4743	4744	4745	4746	4747	4748
4749	4750	4752	4753	4766	4767	4786	4787	4797	4798	4799	4800	4801
4802	4803	4804	4806	4807	4808	4809	4821	4822	4823	4824	4825	4826
4827	4828	4830	4831	4832	4833	4843	4844	4845	4846	4847	4848	4849
4850	4852	4853	4866	4867	4886	4887	4897	4898	4899	4900	4901	4902
4903	4904	4906	4907	4908	4909	4921	4922	4923	4924	4925	4926	4927
4928	4930	4931	4932	4933	4943	4944	4945	4946	4947	4948	4949	4950

4952	4953	4966	4967	5016	5017	5018	5019	5020	5025	5026	5027	5044
5045	5084	5085	5086	5087	5088	5092	5093	5094	5113	5114	5184	5185
5186	5187	5188	5219	5220	5221	5257	5258	5259	5260	5261	5276	5277
5300	5301	5302	5303	5304	5306	5307	5308	5318	5319	5320	5321	5322
5324	5325	5326	5334	5335	5336	5337	5338	5352	5353	5366	5367	5368
5379	5380	5381	5382	5383	5385	5386	5387	5396	5397	5398	5399	5400
5402	5403	5404	5412	5413	5414	5415	5416	5430	5431	5455	5456	5457
5458	5459	5471	5472	5473	5474	5475	5485	5486	5487	5488	5489	5504
5505	5538	5539	5540	5541	5542	5560	5561	5562	5563	5564	5571	5572
5573	5574	5575	5579	5580	5581	5588	5589	5590	5591	5592	5601	5602
5603	5604	5605	5613	5614	5721	5722	5723	5724	5725	5747	5748	5749
5750	5751	5766	5767	5768	5769	5770	5785	5786	5787	5788	5789	5804
5805	5806	5828	5829	5869	5870	5871	5872	5873	5888	5889	5928	5929
5930	5931	5932	5935	5936	5937	5952	5953	5954	5961	5962	6023	6024
6025	6026	6027	6029	6030	6031	6038	6039	6040	6041	6042	6045	6046
6047	6053	6054	6055	6056	6057	6059	6060	6061	6067	6068	6069	6070
6071	6073	6074	6075	6083	6084	6085	6086	6087	6089	6090	6091	6095
6096	6097	6098	6099	6100	6101	6102	6107	6108	6109	6110	6111	6112
6113	6114	6119	6120	6121	6122	6123	6137	6138	6139	6159	6160	6204
6205	6206	6223	6224	6225	6226	6227	6231	6232	6253	6254	6255	6256
6257	6258	6259	6260	6267	6268	6269	6270	6271	6287	6288	6327	6328
6329	6330	6331	6334	6335	6336	6356	6357	6390	6391	6392	6393	6394
6410	6411	6446	6447	6448	6449	6450	6466	6467	6508	6509	6510	6511
6512	6528	6529	6548	6549	6550	6567	6568	6569	6570	6571	6572	6573
6575	6576	6593	6594	6595	6596	6597	6599	6600	6613	6614	6615	6616
6617	6620	6621	6623	6624	6633	6634	6635	6636	6637	6655	6656	6670
6671	6675	6676	6677	6678	6679	6686	6687	6688	6689	6690	6691	6693
6694	6775	6776	6780	6781	6791	6792	6793	6794				
682#	694#											
682#	696#	889	890	904	905	1893	1894	1929	1930	1965	1966	2000
2001	2035	2036	2071	2072	2107	2108	2142	2143	2177	2178	2212	2213
2247	2248	2282	2283	2318	2319	2353	2354	2383	2384	2418	2419	2453
2454	2473	2474	2492	2493	2509	2510	2526	2527	2546	2547	2565	2566
2584	2585	2603	2604	2624	2625	2637	2638	2785	2786	2806	2807	2926
2827	2844	2845	2890	2891	2956	2957	2991	2992	3012	3013	3047	3048
3069	3070	3107	3108	3130	3131	3149	3150	3212	3213	3276	3277	3365
3366	3483	3484	3568	3569	3611	3612	3635	3636	3657	3658	3671	3672
3711	3712	3732	3733	3753	3754	3767	3768	3809	3810	3832	3833	3854
3855	3868	3869	3909	3910	3931	3932	3953	3954	3967	3968	4009	4010
4031	4032	4053	4054	4067	4068	4109	4110	4131	4132	4153	4154	4167
4168	4208	4209	4230	4231	4252	4253	4266	4267	4307	4308	4329	4330
4351	4352	4365	4366	4405	4406	4429	4430	4451	4452	4465	4466	4505
4506	4529	4530	4551	4552	4565	4566	4605	4606	4629	4630	4651	4652
4665	4666	4705	4706	4729	4730	4751	4752	4765	4766	4805	4806	4829
4830	4851	4852	4865	4866	4905	4906	4929	4930	4951	4952	4965	4966
5043	5044	5112	5113	5275	5276	5351	5352	5429	5430	5503	5504	5612
5613	5827	5828	5887	5888	5960	5961	6158	6159	6230	6231	6286	6287
6355	6356	6409	6410	6465	6466	6527	6528	6654	6655	6694	6695	6781
6782												
682#	693#	2861	2862	2902	2903	2966	2967	3022	3023	3080	3081	3159
3160	3222	3223	3287	3288	3376	3377	3495	3496	3584	3585	3687	3688
3783	3784	3884	3885	3983	3984	4083	4084	4183	4184	4282	4283	4381
4382	4481	4482	4581	4582	4681	4682	4781	4782	4881	4882	4979	4980
5058	5059	5151	5152	5288	5289	5364	5365	5441	5442	5519	5520	5626
5627	5840	5841	5899	5900	5972	5973	6172	6173	6243	6244	6299	6300
6371	6372	6425	6426	6481	6482	6542	6543					

SVCSUB= 000000
SVCTAG= 000000

SVCTST= 000000

CZKMAO KMC11-B STATIC PART2
CZKMA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 PAGE 182
CROSS REFERENCE TABLE -- USER SYMBOLS

M 14

SEQ 0181

SV05	004322	1592#	2878	2920	2938	2983	3039	3099	3122	3185	3249	3313	3342	3404
		3426	3466	3514	3535	3551	3603	3627	3649	3703	3724	3745	3800	3823
		3846	3901	3923	3945	4001	4023	4045	4101	4123	4145	4200	4222	4244
		4299	4321	4343	4397	4421	4443	4497	4521	4543	4597	4621	4643	4697
		4721	4743	4797	4821	4843	4897	4921	4943	5016	5084	5184	5257	5300
		5318	5334	5379	5396	5412	5455	5471	5485	5538	5560	5571	5588	5601
		5721	5747	5766	5785	5869	5928	6023	6038	6053	6067	6083	6095	6107
		6119	6223	6253	6267	6327	6390	6446	6508	6569	6593	6613	6633	
SWPAC1	031536	6726#												
SWPAC2	031577	6732#												
SLSYM=	010000	682#	890#	905#	1894#	1930#	1966#	2001#	2036#	2072#	2108#	2143#	2178#	2213#
		2248#	2283#	2319#	2354#	2384#	2419#	2454#	2474#	2493#	2510#	2527#	2547#	2566#
		2585#	2604#	2625#	2638#	2786#	2807#	2827#	2845#	2891#	2957#	2974#	3013#	3029#
		3070#	3086#	3110#	3150#	3213#	3277#	3366#	3484#	3569#	3590#	3614#	3638#	3672#
		3693#	3714#	3735#	3768#	3789#	3812#	3835#	3869#	3890#	3912#	3934#	3968#	3990#
		4012#	4034#	4068#	4090#	4112#	4134#	4168#	4189#	4211#	4233#	4267#	4288#	4310#
		4332#	4366#	4386#	4408#	4432#	4466#	4486#	4508#	4532#	4566#	4586#	4608#	4632#
		4666#	4686#	4708#	4732#	4766#	4786#	4808#	4832#	4866#	4886#	4908#	4932#	4966#
		5044#	5113#	5276#	5352#	5430#	5504#	5613#	5828#	5888#	5961#	6159#	6231#	6287#
		6356#	6410#	6466#	6528#	6655#	6695#	6782#						
TEMP	002402	1105#	5444*	5480*	5526*	5534*								
TESTAD	003322	1431*	1437#											
TFM1	004356	1612#	1883	1919	1955									
TFM2	004402	1616#	1990	2025	2132	2237	2408	2443						
TFM3	004420	1619#	2061	2097										
TFM36	004507	1630#	2461											
TFM4	004445	1623#	2167	2202	2272	2343								
TFM40	004753	1660#	2480											
TFM41	004575	1640#	2497											
TFM42	004664	1650#	2514											
TFM43	005035	1669#	2534											
TFM44	005120	1678#	2553											
TFM45	005157	1684#	2572											
TFM46	005215	1690#	2591											
TFM5	004463	1626#	2308											
TMMC	005301	1699#	2552	2571										
TYPE	002374	1102#	2772*	2777*	2782*	5701								
T\$ARGC=	000001	715#	716#	717#	718#	719#	720#	1861#	1867	1868#	1873	1874#	1879	1880#
		1887	1888#	1892	1897#	1903	1904#	1909	1910#	1915	1916#	1923	1924#	1928
		1933#	1939	1940#	1945	1946#	1951	1952#	1959	1960#	1964	1969#	1975	1976#
		1981	1982#	1987	1988#	1994	1995#	1999	2004#	2010	2011#	2016	2017#	2022
		2023#	2029	2030#	2034	2039#	2045	2046#	2051	2052#	2057	2058#	2065	2066#
		2070	2075#	2081	2082#	2087	2088#	2093	2094#	2101	2102#	2106	2111#	2117
		2118#	2123	2124#	2129	2130#	2136	2137#	2141	2146#	2152	2153#	2158	2159#
		2164	2165#	2171	2172#	2176	2181#	2187	2188#	2193	2194#	2199	2200#	2206
		2207#	2211	2216#	2222	2223#	2228	2229#	2234	2235#	2241	2242#	2246	2251#
		2257	2258#	2263	2264#	2269	2270#	2276	2277#	2281	2286#	2292	2293#	2298
		2299#	2304	2305#	2312	2313#	2317	2322#	2328	2329#	2334	2335#	2340	2341#
		2347	2348#	2352	2359#	2365	2366#	2371	2372#	2377	2378#	2382	2387#	2393
		2394#	2399	2400#	2405	2406#	2412	2413#	2417	2422#	2428	2429#	2434	2435#
		2440	2441#	2447	2448#	2452	2459#	2465	2467#	2471	2479#	2484	2486#	2490
		2497#	2501	2503#	2507	2514#	2518	2520#	2524	2532#	2538	2540#	2544	2551#
		2557	2559#	2563	2570#	2576	2578#	2582	2589#	2595	2597#	2601		
T\$CODE=	011032	6675#	6686#											
T\$ERRN=	000056	682#	2879#	2921#	2939#	2984#	3040#	3100#	3123#	3186#	3250#	3314#	3343#	3405#
		3427#	3467#	3515#	3536#	3552#	3604#	3628#	3650#	3704#	3725#	3746#	3801#	3824#

TSEXCP= 000000
TSFLAG= 000040

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

TSLTNO= 000052
TSNEST= 000000

TSNSO = 000000
TNSI = 000005

3847#	3902#	3924#	3946#	4002#	4024#	4046#	4102#	4124#	4146#	4201#	4223#	4245#
4300#	4322#	4344#	4398#	4422#	4444#	4498#	4522#	4544#	4598#	4622#	4644#	4698#
4722#	4744#	4798#	4822#	4844#	4898#	4922#	4944#	5017#	5085#	5185#	5258#	5301#
5319#	5335#	5380#	5397#	5413#	5456#	5472#	5486#	5539#	5561#	5572#	5589#	5602#
5722#	5748#	5767#	5786#	5870#	5929#	6024#	6039#	6054#	6068#	6084#	6096#	6108#
6120#	6224#	6254#	6268#	6328#	6391#	6447#	6509#	6570#	6594#	6614#	6634#	
6675#	6679	6686#	6691									
2619#	2621	2869#	2886#	2926#	2988#	3044#	3104#	3127#	3190#	3254#	3318#	3347#
3434#	3519#	3608#	3632#	3654#	3708#	3729#	3750#	3806#	3829#	3851#	3906#	3928#
3950#	4006#	4028#	4050#	4106#	4128#	4150#	4205#	4227#	4249#	4304#	4326#	4348#
4402#	4426#	4448#	4502#	4526#	4548#	4602#	4626#	4648#	4702#	4726#	4748#	4802#
4826#	4848#	4902#	4926#	4948#	5025#	5092#	5219#	5306#	5324#	5366#	5385#	5402#
5579#	5804#	5935#	5952#	6029#	6045#	6059#	6073#	6089#	6100#	6112#	6137#	6204#
6258#	6334#											
682#												
6675#	6678	6686#	6690									
682#	6792#											
6675#	6677	6686#	6689									
682#	890	905	1894	1930	1966	2001	2036	2072	2108	2143	2178	2213
2248	2283	2319	2354	2384	2419	2454	2474	2493	2510	2527	2547	2566
2585	2604	2625	2638	2786	2807	2827	2845	2891	2957	3013	3070	3150
3213	3277	3366	3484	3569	3672	3768	3869	3968	4068	4168	4267	4366
4466	4566	4666	4766	4866	4966	5044	5113	5276	5352	5430	5504	5613
5828	5888	5961	6159	6231	6287	6356	6410	6466	6528	6655	6695	6782
6795#												
682#	688#	874#	889#	898#	904#	1860#	1893#	1896#	1929#	1932#	1965#	1968#
2000#	2003#	2035#	2038#	2071#	2074#	2107#	2110#	2142#	2145#	2177#	2180#	2212#
2215#	2247#	2250#	2282#	2285#	2318#	2321#	2353#	2358#	2383#	2386#	2418#	2421#
2453#	2457#	2473#	2477#	2492#	2495#	2509#	2512#	2526#	2530#	2546#	2549#	2565#
2568#	2584#	2587#	2603#	2615#	2624#	2628#	2633#	2635#	2637#	2649#	2785#	2802#
2806#	2821#	2826#	2842#	2844#	2862#	2890#	2903#	2956#	2967#	2974#	2991#	3012#
3023#	3029#	3047#	3069#	3081#	3086#	3107#	3110#	3130#	3149#	3160#	3212#	3223#
3276#	3288#	3365#	3377#	3483#	3496#	3568#	3585#	3590#	3611#	3614#	3635#	3638#
3657#	3671#	3688#	3693#	3711#	3714#	3732#	3735#	3753#	3767#	3784#	3789#	3909#
3812#	3832#	3835#	3854#	3868#	3885#	3890#	3909#	3912#	3931#	3934#	3953#	3967#
3984#	3990#	4009#	4012#	4031#	4034#	4053#	4067#	4084#	4090#	4109#	4112#	4131#
4134#	4153#	4167#	4184#	4189#	4208#	4211#	4230#	4233#	4252#	4266#	4283#	4288#
4307#	4310#	4329#	4332#	4351#	4365#	4382#	4386#	4405#	4408#	4429#	4432#	4451#
4465#	4482#	4486#	4505#	4508#	4529#	4532#	4551#	4565#	4582#	4586#	4605#	4608#
4629#	4632#	4651#	4665#	4682#	4686#	4705#	4708#	4729#	4732#	4751#	4765#	4782#
4786#	4805#	4808#	4829#	4832#	4851#	4865#	4882#	4886#	4905#	4908#	4929#	4932#
4951#	4965#	4980#	5043#	5059#	5112#	5152#	5275#	5289#	5351#	5365#	5429#	5442#
5503#	5520#	5612#	5627#	5827#	5841#	5887#	5900#	5960#	5973#	6158#	6173#	6230#
6244#	6286#	6300#	6355#	6372#	6409#	6426#	6465#	6482#	6527#	6543#	6654#	6670#
6693#	6775#	6780#										
688#												
874#	889	898#	904	1860#	1893	1896#	1929	1932#	1965	1968#	2000	2003#
2035	2038#	2071	2074#	2107	2110#	2142	2145#	2177	2180#	2212	2215#	2247
2250#	2282	2285#	2318	2321#	2353	2358#	2383	2386#	2418	2421#	2453	2457#
2473	2477#	2492	2495#	2509	2512#	2526	2530#	2546	2549#	2565	2568#	2584
2587#	2603	2615#	2624	2628#	2633	2635#	2637	2649#	2785	2802#	2806	2821#
2826	2842#	2844	2862#	2890	2903#	2956	2967#	3012	3023#	3069	3081#	3149
3160#	3212	3223#	3276	3288#	3365	3377#	3483	3496#	3568	3585#	3671	3688#
3767	3784#	3868	3885#	3967	3984#	4067	4084#	4167	4184#	4266	4283#	4365
4382#	4465	4482#	4565	4582#	4665	4682#	4765	4782#	4865	4882#	4965	4980#
5043	5059#	5112	5152#	5275	5289#	5351	5365#	5429	5442#	5503	5520#	5612

CZKMAO KMC11-B STATIC PART2
CZKMA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 PAGE 184
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0183

TSNS2 = 000003

5627#	5827	5841#	5887	5900#	5960	5973#	6158	6173#	6230	6244#	6286	6300#
635	6372#	6409	6426#	6465	6482#	6527	6543#	6654	6670#	6693	6775#	6780
2974#	2991	3029#	3047	3086#	3107	3110#	3130	3590#	3611	3614#	3635	3638#
3657	3693#	3711	3714#	3732	3735#	3753	3789#	3809	3812#	3832	3835#	3854
3890#	3909	3912#	3931	3934#	3953	3990#	4009	4012#	4031	4034#	4053	4090#
4109	4112#	4131	4134#	4153	4189#	4208	4211#	4230	4233#	4252	4288#	4307
4310#	4329	4332#	4351	4386#	4405	4408#	4429	4432#	4451	4486#	4505	4508#
4529	4532#	4551	4586#	4605	4608#	4629	4632#	4651	4686#	4705	4708#	4729
4732#	4751	4786#	4805	4808#	4829	4832#	4851	4886#	4905	4908#	4929	4932#
4951												

TSPTNU= 000000
TSSAVL= 177777
TSSSEGL= 177777

682#	682#	682#	2974#	2989	2991#	2993	3029#	3045	3047#	3049	3086#	3105	3107#	3109
3110#	3128	3130#	3132	3590#	3609	3611#	3613	3614#	3633	3635#	3637	3638#	3655	3657#
3655	3657#	3659	3693#	3709	3711#	3713	3714#	3730	3732#	3734	3735#	3751	3753#	3755
3753#	3755	3789#	3807	3809#	3811	3812#	3830	3832#	3834	3835#	3852	3854#	3856	3890#
3856	3890#	3907	3909#	3911	3912#	3929	3931#	3933	3934#	3951	3953#	3955	3990#	4007
3990#	4007	4009#	4011	4012#	4029	4031#	4033	4034#	4051	4053#	4055	4090#	4107	4109#
4107	4109#	4111	4112#	4129	4131#	4133	4134#	4151	4153#	4155	4189#	4206	4208#	4210
4208#	4210	4211#	4228	4230#	4232	4233#	4250	4252#	4254	4288#	4305	4307#	4309	4310#
4309	4310#	4327	4329#	4331	4332#	4349	4351#	4353	4386#	4403	4405#	4407	4408#	4427
4408#	4427	4429#	4431	4432#	4449	4451#	4453	4486#	4503	4505#	4507	4508#	4527	4529#
4527	4529#	4531	4532#	4549	4551#	4553	4586#	4603	4605#	4607	4608#	4627	4629#	4631
4629#	4631	4632#	4649	4651#	4653	4686#	4703	4705#	4707	4708#	4727	4729#	4731	4732#
4731	4732#	4749	4751#	4753	4786#	4803	4805#	4807	4808#	4827	4829#	4831	4832#	4849
4832#	4849	4851#	4853	4886#	4903	4905#	4907	4908#	4927	4929#	4931	4932#	4949	4951#
4949	4951#	4953												

TSSEK0= 010002

2974#	2989	2991	3029#	3045	3047	3086#	3105	3107	3110#	3128	3130	3590#
3609	3611	3614#	3633	3635	3638#	3655	3657	3693#	3709	3711	3714#	3730
3732	3735#	3751	3753	3789#	3807	3809	3812#	3830	3832	3835#	3852	3854#
3890#	3907	3909	3912#	3929	3931	3934#	3951	3953	3990#	4007	4009	4012#
4029	4031	4034#	4051	4053	4090#	4107	4109	4112#	4129	4131	4134#	4151
4153	4189#	4206	4208	4211#	4228	4230	4233#	4250	4252	4288#	4305	4307#
4310#	4327	4329	4332#	4349	4351	4386#	4403	4405	4408#	4427	4429	4432#
4449	4451	4486#	4503	4505	4508#	4527	4529	4532#	4549	4551	4586#	4603
4605	4608#	4627	4629	4632#	4649	4651	4686#	4703	4705	4708#	4727	4729#
4732#	4749	4751	4786#	4803	4805	4808#	4827	4829	4832#	4849	4851	4886#
4903	4905	4908#	4927	4929	4932#	4949	4951					

TSSUBN= 000000

682#	2861#	2902#	2966#	3022#	3080#	3159#	3222#	3287#	3376#	3495#	3584#	3687#
3783#	3884#	3983#	4083#	4183#	4282#	4381#	4481#	4581#	4681#	4781#	4881#	4979#
5058#	5151#	5288#	5364#	5441#	5519#	5626#	5840#	5899#	5972#	6172#	6243#	6299#
6371#	6425#	6481#	6542#									

TSTAGL= 177777
TSTAGN= 010116

682#	874#	898#	1860#	1896#	1932#	1968#	2003#	2038#	2074#	2110#	2145#	2180#
2215#	2250#	2285#	2321#	2358#	2386#	2421#	2457#	2477#	2495#	2512#	2530#	2549#
2568#	2587#	2615#	2628#	2635#	2649#	2802#	2821#	2842#	2862#	2903#	2967#	3023#
3081#	3160#	3223#	3288#	3377#	3496#	3585#	3688#	3784#	3885#	3984#	4084#	4184#
4283#	4382#	4482#	4582#	4682#	4782#	4882#	4980#	5059#	5152#	5289#	5365#	5442#
5520#	5627#	5841#	5900#	5973#	6173#	6244#	6300#	6372#	6426#	6482#	6543#	6670#
6775#												

TSTEMP= 000005

816#	817#	818#	819#	820#	821#	822#	823#	824#	825#	826#	827#	828#
829#	830#	831#	832#	833#	834#	835#	836#	837#	838#	839#	840#	841#
842#	843#	844#	845#	846#	847#	848#	849#	850#	851#	852#	853#	854#
855#	856#	857#	858#	889#	904#	1893#	1929#	1965#	2000#	2035#	2071#	2107#
2142#	2177#	2212#	2247#	2282#	2318#	2353#	2383#	2418#	2453#	2473#	2492#	2509#
2526#	2546#	2565#	2584#	2603#	2619#	2620	2624#	2633#	2637#	2785#	2806#	2826#

2844#	2869#	2870	2886#	2887	2890#	2926#	2927	2956#	2988#	2989#	2991#	3012#
3044#	3045#	3047#	3069#	3104#	3105#	3107#	3127#	3128#	3130#	3149#	3190#	3191
3212#	3254#	3255	3276#	3318#	3319	3347#	3348	3365#	3434#	3435	3483#	3519#
3520	3568#	3608#	3609#	3611#	3632#	3633#	3635#	3654#	3655#	3657#	3671#	3708#
3709#	3711#	3729#	3730#	3732#	3750#	3751#	3753#	3767#	3806#	3807#	3809#	3829#
3830#	3832#	3851#	3852#	3854#	3868#	3906#	3907#	3909#	3928#	3929#	3931#	3950#
3951#	3953#	3967#	4006#	4007#	4009#	4028#	4029#	4031#	4050#	4051#	4053#	4067#
4106#	4107#	4109#	4128#	4129#	4131#	4150#	4151#	4153#	4167#	4205#	4206#	4208#
4227#	4228#	4230#	4249#	4250#	4252#	4266#	4304#	4305#	4307#	4326#	4327#	4329#
4348#	4349#	4351#	4365#	4402#	4403#	4405#	4426#	4427#	4429#	4448#	4449#	4451#
4465#	4502#	4503#	4505#	4526#	4527#	4529#	4548#	4549#	4551#	4565#	4602#	4603#
4605#	4626#	4627#	4629#	4648#	4649#	4651#	4665#	4702#	4703#	4705#	4726#	4727#
4729#	4748#	4749#	4751#	4765#	4802#	4803#	4805#	4826#	4827#	4829#	4848#	4849#
4851#	4865#	4902#	4903#	4905#	4926#	4927#	4929#	4948#	4949#	4951#	4965#	5025#
5026	5043#	5092#	5093	5112#	5219#	5220	5275#	5306#	5307	5324#	5325	5351#
5366#	5367	5385#	5386	5402#	5403	5429#	5503#	5579#	5580	5612#	5804#	5805
5827#	5887#	5935#	5936	5952#	5953	5960#	6029#	6030	6045#	6046	6059#	6060
6073#	6074	6089#	6090	6100#	6101	6112#	6113	6137#	6138	6158#	6204#	6205
6230#	6258#	6259	6286#	6334#	6335	6355#	6409#	6465#	6527#	6654#	6675#	6686#
6693#	6780#											
682#	2854	2858	2861#	2894	2899	2902#	2959	2963	2966#	3015	3019	3022#
3072	3077	3080#	3152	3156	3159#	3215	3219	3222#	3279	3284	3287#	3368
3373	3376#	3486	3492	3495#	3571	3581	3584#	3674	3684	3687#	3770	3780
3783#	3871	3881	3884#	3970	3980	3983#	4070	4080	4083#	4170	4180	4183#
4269	4279	4282#	4368	4378	4381#	4468	4478	4481#	4568	4578	4581#	4668
4678	4681#	4768	4778	4781#	4868	4878	4881#	4968	4976	4979#	5046	5055
5058#	5115	5148	5151#	5278	5285	5288#	5354	5361	5364#	5432	5438	5441#
5506	5516	5519#	5615	5623	5626#	5830	5837	5840#	5890	5896	5899#	5963
5969	5972#	6161	6169	6172#	6233	6240	6243#	6289	6296	6299#	6358	6368
6371#	6412	6422	6425#	6468	6478	6481#	6530	6539	6542#	6795		
682#	1866	1872	1878	1886	1891	1894	1902	1908	1914	1922	1927	1930
1938	1944	1950	1958	1963	1966	1974	1980	1986	1993	1998	2001	2009
2015	2021	2028	2033	2036	2044	2050	2056	2064	2069	2072	2080	2086
2092	2100	2105	2108	2116	2122	2128	2135	2140	2143	2151	2157	2163
2170	2175	2178	2186	2192	2198	2205	2210	2213	2221	2227	2233	2240
2245	2248	2256	2262	2268	2275	2280	2283	2291	2297	2303	2311	2316
2319	2327	2333	2339	2346	2351	2354	2364	2370	2376	2381	2384	2392
2398	2404	2411	2416	2419	2427	2433	2439	2446	2451	2454	2464	2470
2474	2483	2489	2493	2500	2506	2510	2517	2523	2527	2537	2543	2547
2556	2562	2566	2575	2581	2585	2594	2600	2604	2625	2638	2666	2672
2677	2684	2699	2786	2807	2824	2827	2845	2869	2878	2886	2891	2920
2926	2938	2957	2974	2983	2988	2992	3013	3029	3039	3044	3048	3070
3086	3099	3104	3108	3110	3122	3127	3131	3150	3185	3190	3213	3249
3254	3277	3313	3318	3342	3347	3366	3404	3426	3434	3466	3484	3514
3519	3535	3551	3569	3590	3603	3608	3612	3614	3627	3632	3636	3638
3649	3654	3658	3672	3693	3703	3708	3712	3714	3724	3729	3733	3735
3745	3750	3754	3768	3789	3800	3806	3810	3812	3823	3829	3833	3835
3846	3851	3855	3869	3890	3901	3906	3910	3912	3923	3928	3932	3934
3945	3950	3954	3968	3990	4001	4006	4010	4012	4023	4028	4032	4034
4045	4050	4054	4068	4090	4101	4106	4110	4112	4123	4128	4132	4134
4145	4150	4154	4168	4189	4200	4205	4209	4211	4222	4227	4231	4233
4244	4249	4253	4267	4288	4299	4304	4308	4310	4321	4326	4330	4332
4343	4348	4352	4366	4386	4397	4402	4406	4408	4421	4426	4430	4432
4443	4448	4452	4466	4486	4497	4502	4506	4508	4521	4526	4530	4532
4543	4548	4552	4566	4586	4597	4602	4606	4608	4621	4626	4630	4632
4643	4648	4652	4666	4686	4697	4702	4706	4708	4721	4726	4730	4732

T\$TEST= 000052

T\$TSTM= 177777

CZKMAO KMC11-B STATIC PART2
CZKMA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 PAGE 186
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0185

4743	4748	4752	4766	4786	4797	4802	4806	4808	4821	4826	4830	4832
4843	4848	4852	4866	4886	4897	4902	4906	4908	4921	4926	4930	4932
4943	4948	4952	4966	5016	5025	5044	5084	5092	5113	5184	5219	5257
5276	5300	5306	5318	5324	5334	5352	5366	5379	5385	5396	5402	5412
5430	5455	5471	5485	5504	5538	5560	5571	5579	5588	5601	5613	5721
5747	5766	5785	5804	5828	5869	5888	5928	5935	5952	5961	6023	6029
6038	6045	6053	6059	6067	6073	6073	6089	6095	6100	6107	6112	6119
6137	6159	6204	6223	6231	6253	6258	6267	6287	6327	6334	6356	6390
6410	6446	6466	6508	6528	6549	6567	6569	6575	6593	6599	6613	6620
6623	6633	6655										
682#	2862#	2903#	2967#	3023#	3081#	3160#	3223#	3288#	3377#	3496#	3585#	3688#
3784#	3885#	3984#	4084#	4184#	4283#	4382#	4482#	4582#	4682#	4782#	4882#	4980#
5059#	5152#	5289#	5365#	5442#	5520#	5627#	5841#	5900#	5973#	6173#	6244#	6300#
6372#	6426#	6482#	6543#									
2842#	2844											
2635#	2637											
2802#	2806											
2821#	2826											
6670#	6694											
874#	889											
2649#	2785											
1860#	1893	1896#	1929	1932#	1965	1968#	2000	2003#	2035	2038#	2071	2074#
2107	2110#	2142	2145#	2177	2180#	2212	2215#	2247	2250#	2282	2285#	2318
2321#	2353	2358#	2383	2386#	2418	2421#	2453	2457#	2473	2477#	2492	2495#
2509	2512#	2526	2530#	2546	2549#	2565	2568#	2584	2587#	2603		
2628#												
2615#	2619	2624										
2974#	2988	2991#	3029#	3044	3047#	3086#	3104	3107#	3110#	3127	3130#	3590#
3608	3611#	3614#	3632	3635#	3638#	3654	3657#	3693#	3708	3711#	3714#	3729
3732#	3735#	3750	3753#	3789#	3806	3809#	3812#	3829	3832#	3835#	3851	3854#
3890#	3906	3909#	3912#	3928	3931#	3934#	3950	3953#	3990#	4006	4009#	4012#
4028	4031#	4034#	4050	4053#	4090#	4106	4109#	4112#	4128	4131#	4134#	4150
4153#	4189#	4205	4208#	4211#	4227	4230#	4233#	4249	4252#	4288#	4304	4307#
4310#	4326	4329#	4332#	4348	4351#	4386#	4402	4405#	4408#	4426	4429#	4432#
4448	4451#	4486#	4502	4505#	4508#	4526	4529#	4532#	4548	4551#	4586#	4602
4605#	4608#	4626	4629#	4632#	4648	4651#	4686#	4702	4705#	4708#	4726	4729#
4732#	4748	4751#	4786#	4802	4805#	4808#	4826	4829#	4832#	4848	4851#	4886#
4902	4905#	4908#	4926	4929#	4932#	4948	4951#					
6775#	6781											
898#	904											
2862#	2869	2886	2890	2903#	2926	2956	2967#	3012	3023#	3069	3081#	3149
3100#	3190	3212	3223#	3254	3276	3288#	3318	3347	3365	3377#	3434	3483
3496#	3519	3568	3585#	3671	3688#	3767	3784#	3868	3885#	3967	3984#	4067
4084#	4167	4184#	4266	4283#	4365	4382#	4465	4482#	4565	4582#	4665	4682#
4765	4782#	4865	4882#	4965	4980#	5025	5043	5059#	5092	5112	5152#	5219
5275	5289#	5306	5324	5351	5365#	5366	5385	5402	5429	5442#	5503	5520#
5579	5612	5627#	5804	5827	5841#	5887	5900#	5935	5952	5960	5973#	6029
6045	6059	6073	6089	6100	6112	6137	6158	6173#	6204	6230	6244#	6258
6286	6300#	6334	6355	6372#	6409	6426#	6465	6482#	6527	6543#	6654	
816	2861#											
825	3495#											
826	3584#											
827	3687#											
828	3783#											
829	3884#											
830	3983#											

TSTSTS= 000001

TSSAU = 010041
TSSAUT= 010035
TSSCLE= 010037
TSSDU = 010040
TSSHAR= 010114
TSSHW = 010000
TSSINI= 010036
TSSMSG= 010032

TSSPRO= 010034
TSSRPT= 010033
TSSSEG= 010002

TSSSOF= 010115
TSSSW = 010001
TSSTES= 010113

T1 013116 G
T10 015650 G
T11 016134 G
T12 016420 G
T13 016664 G
T14 017144 G
T15 017424 G

CZKMAO KMC11-B STATIC PART2
CZKMA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 PAGE 188
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0187

\$REG2	002366	1099#	1595*	1882	1918	1952	2094	2166	2271					
\$REG3	002364	1098#	1594*											
\$REG4	002362	1097#	1593*	1881	1917	1953	1988	2023	2059	2095	2130	2165	2200	2235
		2270	2305	2341	2406	2441								
\$REG5	002360	1096#	1592*	1954	1989	2024	2060	2096	2131	2201	2236	2306	2342	2407
		2442												
\$TEMPO	002404	1106#												
\$TMPO	002406	1107#	5445*	5481*	5482	5527*	5557	5564						
.	= 032002	673#	1088#	1089#	1090#	1091#	1123#	1169#	1200#	1438#	2620	2870	2887	2927
		2989	3045	3105	3128	3191	3255	3319	3348	3435	3520	3609	3633	3655
		3709	3730	3751	3807	3830	3852	3907	3929	3951	4007	4029	4051	4107
		4129	4151	4206	4228	4250	4305	4327	4349	4403	4427	4449	4503	4527
		4549	4603	4627	4649	4703	4727	4749	4803	4827	4849	4903	4927	4949
		5026	5093	5220	5307	5325	5367	5386	5403	5580	5805	5809#	5813#	5936
		5953	6030	6046	6060	6074	6090	6101	6113	6138	6205	6259	6335	6755#
.MSTCL	003254	1417#	2905	2969	3025	3084	3163	3226	3291	3380	3499	3589	3691	3787
		3888	3987	4087	4186	4286	4384	4388	4484	4488	4584	4684	4784	4884
		4982	5061	5154	5290	5369	5444	5523	5537	5587	5600	5610	5629	5843
		5903	5977	6175	6246	6302	6376	6428	6486	6545				
.REGT	003640	1447#												
.ROMCL	003634	1324	1444#	1472	1474	1476	1484	1491	1498	1505	1512	1514	1516	1525
		1536	1538	2910	2912	2914	2930	2932	3170	3172	3175	3177	3179	3234
		3236	3239	3241	3243	3298	3300	3303	3305	3307	3330	3332	3334	3336
		3384	3386	3388	3394	3396	3412	3415	3453	3455	3501	3504	3523	3528
		3540	3542	3544	4986	4988	4992	5007	5009	5068	5070	5077	5165	5167
		5173	5175	5243	5245	5293	5310	5328	5372	5389	5406	5448	5477	5634
		5647	5650	5653	5666	5669	5672	5678	5680	5683	5686	5689	5697	5699
		5714	5742	5758	5761	5775	5777	5780	5844	5846	5848	5850	5854	5857
		6317	6320	6381	6435	6441	6488	6491	6502	6552	6555	6557	6560	6582
		6585	6604	6606	6626									
.SROMC	003670	1453#	1478	1518	1527	3398	3409	3449	3451	3595	3597	3619	3621	3641
		3643	3695	3697	3716	3718	3737	3739	3792	3794	3815	3817	3838	3840
		3893	3895	3915	3917	3937	3939	3993	3995	4015	4017	4037	4039	4093
		4095	4115	4117	4137	4139	4192	4194	4214	4216	4236	4238	4291	4293
		4313	4315	4335	4337	4389	4391	4413	4415	4435	4437	4489	4491	4513
		4515	4535	4537	4589	4591	4613	4615	4635	4637	4689	4691	4713	4715
		4735	4737	4789	4791	4813	4815	4835	4837	4889	4891	4913	4915	4935
		4937	5072	5169	5177	5239	5241	6378	6493					

CZKMAO KMC11-B STATIC PART2
CZKMA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 PAGE 190
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0188

BADHEA	1248#	2853	2857	2893	2898	2958	2962	3014	3018	3071	3076	3151	3155	3214	3218
	3278	3283	3367	3372	3485	3491	3570	3580	3673	3683	3769	3779	3870	3880	3969
	3979	4069	4079	4169	4179	4268	4278	4367	4377	4467	4477	4567	4577	4667	4677
	4767	4777	4867	4877	4967	4975	5045	5054	5114	5147	5277	5284	5353	5360	5431
	5437	5505	5515	5614	5622	5829	5836	5889	5895	5962	5968	6160	6168	6232	6239
	6288	6295	6357	6367	6411	6421	6467	6477	6529	6538					
BCOMPL	2667	2673	2678												
BGNAU	2841														
BGNAUT	2634														
BGNCLN	2801														
BGNDU	2820														
BGNHRD	6669														
BGNHW	873														
BGNINI	2648														
BGNMOD	687														
BGNMSG	1860	1896	1932	1968	2003	2038	2074	2110	2145	2180	2215	2250	2285	2321	2358
	2386	2421	2456	2476	2494	2511	2529	2548	2567	2586					
BGNPRO	2627														
BGNRPT	2614														
BGNSEG	2973	3028	3085	3109	3589	3613	3637	3692	3713	3734	3788	3811	3834	3889	3911
	3933	3989	4011	4033	4089	4111	4133	4188	4210	4232	4287	4309	4331	4385	4407
	4431	4485	4507	4531	4585	4607	4631	4685	4707	4731	4785	4807	4831	4885	4907
	4931														
BGNSFT	6774														
BGNSW	897														
BGNTST	2860	2901	2965	3021	3079	3158	3221	3286	3375	3494	3583	3686	3782	3883	3982
	4082	4182	4281	4380	4480	4580	4680	4780	4880	4978	5057	5150	5287	5363	5440
	5518	5625	5839	5898	5971	6171	6242	6298	6370	6424	6480	6541			
BNCOMP	2685	2701													
BRESET	2823	6566	6622												
CHKREG	1304#	2944	3000	3057	3137	3200	3264	3353	3471	3556	3659	3755	3856	3955	4055
	4155	4254	4353	4453	4553	4653	4753	4853	4953	5031	5100	5263	5339	5417	5491
	5792	5815	5875	5939	6125	6192	6274	6343	6397	6453	6515	6642			
CKLOOP	6574	6598	6619												
CLRMAR	1277#	3385	3395												
DESCRI	1193														
DEVTYP	1189														
DISPAT	813														
ED\$CAL	1243#	2854	2858	2894	2899	2959	2963	3015	3019	3072	3077	3152	3156	3215	3219
	3279	3284	3368	3373	3486	3492	3571	3581	3674	3684	3770	3780	3871	3881	3970
	3980	4070	4080	4170	4180	4269	4279	4368	4378	4468	4478	4568	4578	4668	4678
	4768	4778	4868	4878	4968	4976	5046	5055	5115	5148	5278	5285	5354	5361	5432
	5438	5506	5516	5615	5623	5830	5837	5890	5896	5963	5969	6161	6169	6233	6240
	6289	6296	6358	6368	6412	6422	6468	6478	6530	6539					
ENDAU	2843														
ENDAUT	2636														
ENDCLN	2805														
ENDDU	2825														
ENDHRD	6692														
ENDHW	888														
ENDINI	2784														
ENDMSG	1893	1929	1965	2000	2035	2071	2107	2142	2177	2212	2247	2282	2318	2353	2383
	2418	2453	2472	2491	2508	2525	2545	2564	2583	2602					
ENDPRO	2632														
ENDRPT	2623														
ENDSEG	2990	3046	3106	3129	3610	3634	3656	3710	3731	3752	3808	3831	3853	3908	3930

MSCOUN	1861#	1868#	1874#	1880#	1888#	1897#	1904#	1910#	1916#	1924#	1933#	1940#	1946#	1952#	1960#
	1969#	1976#	1982#	1988#	1995#	2004#	2011#	2017#	2023#	2030#	2039#	2046#	2052#	2058#	2066#
	2075#	2082#	2088#	2094#	2102#	2111#	2118#	2124#	2130#	2137#	2146#	2153#	2159#	2165#	2172#
	2181#	2188#	2194#	2200#	2207#	2216#	2223#	2229#	2235#	2242#	2251#	2258#	2264#	2270#	2277#
	2286#	2293#	2299#	2305#	2313#	2322#	2329#	2335#	2341#	2348#	2359#	2366#	2372#	2378#	2387#
	2394#	2400#	2406#	2413#	2422#	2429#	2435#	2441#	2448#	2459#	2467#	2479#	2486#	2497#	2503#
	2514#	2520#	2532#	2540#	2551#	2559#	2570#	2578#	2589#	2597#					
MSDATA	714#	723	725	727	729	731	733	735	737	739	741	743	745	747	749
	751	753#	755	757	760	763	765	767	769	771	773	775	777	779	781
	783	785	787	789	791	793	795	797	1190#	1194#					
MSDECR	889#	904#	1893#	1929#	1965#	2000#	2035#	2071#	2107#	2142#	2177#	2212#	2247#	2282#	2318#
	2353#	2383#	2418#	2453#	2473#	2492#	2509#	2526#	2546#	2565#	2584#	2603#	2624#	2633#	2637#
	2785#	2806#	2826#	2844#	2890#	2956#	2991#	3012#	3047#	3069#	3107#	3130#	3149#	3212#	3276#
	3365#	3483#	3568#	3611#	3635#	3657#	3671#	3711#	3732#	3753#	3767#	3809#	3832#	3854#	3868#
	3909#	3931#	3953#	3967#	4009#	4031#	4053#	4067#	4109#	4131#	4153#	4167#	4208#	4230#	4252#
	4266#	4307#	4329#	4351#	4365#	4405#	4429#	4451#	4465#	4505#	4529#	4551#	4565#	4605#	4629#
	4651#	4665#	4705#	4729#	4751#	4765#	4805#	4829#	4851#	4865#	4905#	4929#	4951#	4965#	5043#
	5112#	5275#	5351#	5429#	5503#	5612#	5827#	5887#	5960#	6158#	6230#	6286#	6355#	6409#	6465#
	6527#	6654#	6693#	6780#											
MSDEFA	6675#	6686#													
MSENDE	889#	904#	1893#	1929#	1965#	2000#	2035#	2071#	2107#	2142#	2177#	2212#	2247#	2282#	2318#
	2353#	2383#	2418#	2453#	2473#	2492#	2509#	2526#	2546#	2565#	2584#	2603#	2624#	2637#	2785#
	2806#	2826#	2844#	2890#	2956#	2991#	3012#	3047#	3069#	3107#	3130#	3149#	3212#	3276#	3365#
	3483#	3568#	3611#	3635#	3657#	3671#	3711#	3732#	3753#	3767#	3809#	3832#	3854#	3868#	3909#
	3931#	3953#	3967#	4009#	4031#	4053#	4067#	4109#	4131#	4153#	4167#	4208#	4230#	4252#	4266#
	4307#	4329#	4351#	4365#	4405#	4429#	4451#	4465#	4505#	4529#	4551#	4565#	4605#	4629#	4651#
	4665#	4705#	4729#	4751#	4765#	4805#	4829#	4851#	4865#	4905#	4929#	4951#	4965#	5043#	5112#
	5275#	5351#	5429#	5503#	5612#	5827#	5887#	5960#	6158#	6230#	6286#	6355#	6409#	6465#	6527#
	6654#	6693#	6780#												
MSERRI	2878#	2920#	2938#	2983#	3039#	3099#	3122#	3185#	3249#	3313#	3342#	3404#	3426#	3466#	3514#
	3535#	3551#	3603#	3627#	3649#	3703#	3724#	3745#	3800#	3823#	3846#	3901#	3923#	3945#	4001#
	4023#	4045#	4101#	4123#	4145#	4200#	4222#	4244#	4299#	4321#	4343#	4397#	4421#	4443#	4497#
	4521#	4543#	4597#	4621#	4643#	4697#	4721#	4743#	4797#	4821#	4843#	4897#	4921#	4943#	5016#
	5084#	5184#	5257#	5300#	5318#	5334#	5379#	5396#	5412#	5455#	5471#	5485#	5538#	5560#	5571#
	5588#	5601#	5721#	5747#	5766#	5785#	5869#	5928#	6023#	6038#	6053#	6067#	6083#	6095#	6107#
	6119#	6223#	6253#	6267#	6327#	6390#	6446#	6508#	6569#	6593#	6613#	6633#			
MSESCA	2869#	2870	2886#	2887	2926#	2927	2988#	3044#	3104#	3127#	3190#	3191	3254#	3255	3318#
	3319	3347#	3348	3434#	3435	3519#	3520	3608#	3632#	3654#	3708#	3729#	3750#	3806#	3829#
	3851#	3906#	3928#	3950#	4006#	4028#	4050#	4106#	4128#	4150#	4205#	4227#	4249#	4304#	4326#
	4348#	4402#	4426#	4448#	4502#	4526#	4548#	4602#	4626#	4648#	4702#	4726#	4748#	4802#	4826#
	4848#	4902#	4926#	4948#	5025#	5026	5092#	5093	5219#	5220	5306#	5307	5324#	5325	5385#
	5386	5402#	5403	5935#	5936	6029#	6030	6045#	6046	6059#	6060	6073#	6074	6089#	6090
	6100#	6101	6112#	6113	6258#	6259	6334#	6335							
MSESCS	2869#	2886#	2926#	2988#	2989	3044#	3045	3104#	3105	3127#	3128	3190#	3254#	3318#	3347#
	3434#	3519#	3608#	3609	3632#	3633	3654#	3655	3708#	3709	3729#	3730	3750#	3751	3806#
	3807	3829#	3830	3851#	3852	3906#	3907	3928#	3929	3950#	3951	4006#	4007	4028#	4029
	4050#	4051	4106#	4107	4128#	4129	4150#	4151	4205#	4206	4227#	4228	4249#	4250	4304#
	4305	4326#	4327	4348#	4349	4402#	4403	4426#	4427	4448#	4449	4502#	4503	4526#	4527
	4548#	4549	4602#	4603	4626#	4627	4648#	4649	4702#	4703	4726#	4727	4748#	4749	4802#
	4803	4826#	4827	4848#	4849	4902#	4903	4926#	4927	4948#	4949	5025#	5092#	5219#	5306#
	5324#	5385#	5402#	5935#	6029#	6045#	6059#	6073#	6089#	6100#	6112#	6258#	6334#		
MSEXCP	6675#	6686#													
MSEXIT	2619#	5366#	5367	5579#	5580	5804#	5805	5952#	5953	6137#	6138	6204#	6205		
MSEXSE	2619#	5366#	5579#	5804#	5952#	6137#	6204#								
MSEXTJ	2619#	2620	5366#	5579#	5804#	5952#	6137#	6204#							
MSGEN	688#	714#	723#	725#	727#	729#	731#	733#	735#	737#	739#	741#	743#	745#	747#

	749#	751#	753#	755#	757#	760#	763#	765#	767#	769#	771#	773#	775#	777#	779#
	781#	783#	785#	787#	789#	791#	793#	795#	797#	815#	875#	876#	889#	899#	900#
	904#	1190#	1194#	1860#	1893#	1896#	1929#	1932#	1965#	1968#	2000#	2003#	2035#	2038#	2071#
	2074#	2107#	2110#	2142#	2145#	2177#	2180#	2212#	2215#	2247#	2250#	2282#	2285#	2318#	2321#
	2353#	2358#	2383#	2386#	2418#	2421#	2453#	2457#	2473#	2477#	2492#	2495#	2509#	2512#	2526#
	2530#	2546#	2549#	2565#	2568#	2584#	2587#	2603#	2615#	2624#	2628#	2635#	2637#	2649#	2785#
	2802#	2806#	2821#	2826#	2842#	2844#	2861#	2890#	2902#	2956#	2966#	2991#	3012#	3022#	3047#
	3069#	3080#	3107#	3130#	3149#	3159#	3212#	3222#	3276#	3287#	3365#	3376#	3483#	3495#	3568#
	3584#	3611#	3635#	3657#	3671#	3687#	3711#	3732#	3753#	3767#	3783#	3809#	3832#	3854#	3868#
	3884#	3909#	3931#	3953#	3967#	3983#	4009#	4031#	4053#	4067#	4083#	4109#	4131#	4153#	4167#
	4183#	4208#	4230#	4252#	4266#	4282#	4307#	4329#	4351#	4365#	4381#	4405#	4429#	4451#	4465#
	4481#	4505#	4529#	4551#	4565#	4581#	4605#	4629#	4651#	4665#	4681#	4705#	4729#	4751#	4765#
	4781#	4805#	4829#	4851#	4865#	4881#	4905#	4929#	4951#	4965#	4979#	5043#	5058#	5112#	5151#
	5275#	5288#	5351#	5364#	5429#	5441#	5503#	5519#	5612#	5626#	5827#	5840#	5887#	5899#	5960#
	5972#	6158#	6172#	6230#	6243#	6286#	6299#	6355#	6371#	6409#	6425#	6465#	6481#	6527#	6542#
MSGETS	6654#	6671#	6694#	6776#	6781#	6794#									
	889#	904#	1893#	1929#	1965#	2000#	2035#	2071#	2107#	2142#	2177#	2212#	2247#	2282#	2318#
	2353#	2383#	2418#	2453#	2473#	2492#	2509#	2526#	2546#	2565#	2584#	2603#	2624#	2633#	2637#
	2785#	2806#	2826#	2844#	2890#	2956#	2989#	2991#	3012#	3045#	3047#	3069#	3105#	3107#	3128#
	3130#	3149#	3212#	3276#	3365#	3483#	3568#	3609#	3611#	3633#	3635#	3655#	3657#	3671#	3709#
	3711#	3730#	3732#	3751#	3753#	3767#	3807#	3809#	3830#	3832#	3852#	3854#	3868#	3907#	3909#
	3929#	3931#	3951#	3953#	3967#	4007#	4009#	4029#	4031#	4051#	4053#	4067#	4107#	4109#	4129#
	4131#	4151#	4153#	4167#	4206#	4208#	4228#	4230#	4250#	4252#	4266#	4305#	4307#	4327#	4329#
	4349#	4351#	4365#	4403#	4405#	4427#	4429#	4449#	4451#	4465#	4503#	4505#	4527#	4529#	4549#
	4551#	4565#	4603#	4605#	4627#	4629#	4649#	4651#	4665#	4703#	4705#	4727#	4729#	4749#	4751#
	4765#	4803#	4805#	4827#	4829#	4849#	4851#	4865#	4903#	4905#	4927#	4929#	4949#	4951#	4965#
	5043#	5112#	5275#	5351#	5429#	5503#	5612#	5827#	5827#	5960#	6158#	6230#	6286#	6355#	6409#
MSGETT	6465#	6527#	6654#	6693#	6780#										
	2619#	2869#	2886#	2926#	2988#	2989	3044#	3045	3104#	3105	3127#	3128	3190#	3254#	3318#
	3347#	3434#	3519#	3608#	3609	3632#	3633	3654#	3655	3708#	3709	3729#	3730	3750#	3751
	3806#	3807	3829#	3830	3851#	3852	3906#	3907	3928#	3929	3950#	3951	4006#	4007	4028#
	4029	4050#	4051	4106#	4107	4128#	4129	4150#	4151	4205#	4206	4227#	4228	4249#	4250
	4304#	4305	4326#	4327	4348#	4349	4402#	4403	4426#	4427	4448#	4449	4502#	4503	4526#
	4527	4548#	4549	4602#	4603	4626#	4627	4648#	4649	4702#	4703	4726#	4727	4748#	4749
	4802#	4803	4826#	4827	4848#	4849	4902#	4903	4926#	4927	4948#	4949	5025#	5092#	5219#
	5306#	5324#	5366#	5385#	5402#	5579#	5804#	5935#	5952#	6029#	6045#	6059#	6073#	6089#	6100#
MSGNGB	6112#	6137#	6204#	6258#	6334#										
	688#	714#	723#	725#	727#	729#	731#	733#	735#	737#	739#	741#	743#	745#	747#
	749#	751#	753#	755#	757#	760#	763#	765#	767#	769#	771#	773#	775#	777#	779#
	781#	783#	785#	787#	789#	791#	793#	795#	797#	814#	815	874#	875	876	898#
	899	900	1190#	1194#	1860#	1896#	1932#	1968#	2003#	2038#	2074#	2110#	2145#	2180#	2215#
	2250#	2285#	2321#	2358#	2386#	2421#	2457#	2477#	2495#	2512#	2530#	2549#	2568#	2587#	2615#
MSGNIN	2628#	2635#	2649#	2802#	2821#	2842#	6670#	6671	6775#	6776	6791#	6794			
	714#	715	716	717	718	719	720#	721#	722#	723#	724	725#	726	727#	728
	729#	730	731#	732	733#	734	735#	736	737#	738	739#	740	741#	742	743#
	744	745#	746	747#	748	749#	750	751#	752	753#	754	755#	756	757#	758
	759	760#	761	762#	763#	764	765#	766	767#	768	769#	770	771#	772	773#
	774	775#	776	777#	778	779#	780	781#	782	783#	784	785#	786	787#	788
	789#	790	791#	792	793#	794	795#	796	797#	798	814#	816#	817#	818#	819#
	820#	821#	822#	823#	824#	825#	826#	827#	828#	829#	830#	831#	832#	833#	834#
	835#	836#	837#	838#	839#	840#	841#	842#	843#	844#	845#	846#	847#	848#	849#
	850#	851#	852#	853#	854#	855#	856#	857#	874#	898#	1190#	1191	1192	1194#	1195
	1200	1861#	1862#	1863#	1864#	1865	1866#	1867	1868#	1869#	1870#	1871	1872#	1873	1874#
	1875#	1876#	1877	1878#	1879	1880#	1881#	1882#	1883#	1884#	1885	1886#	1887	1888#	1889#
	1890	1891#	1892	1894#	1897#	1898#	1899#	1900#	1901	1902#	1903	1904#	1905#	1906#	1907
	1908#	1909	1910#	1911#	1912#	1913	1914#	1915	1916#	1917#	1918#	1919#	1920#	1921	1922#

1923	1924#	1925#	1926	1927#	1928	1930#	1933#	1934#	1935#	1936#	1937	1938#	1939	1940#
1941#	1942#	1943	1944#	1945	1946#	1947#	1948#	1949	1950#	1951	1952#	1953#	1954#	1955#
1956#	1957	1958#	1959	1960#	1961#	1962	1963#	1964	1966#	1969#	1970#	1971#	1972#	1973
1974#	1975	1976#	1977#	1978#	1979	1980#	1981	1982#	1983#	1984#	1985	1986#	1987	1988#
1989#	1990#	1991#	1992	1993#	1994	1995#	1996#	1997	1998#	1999	2001#	2004#	2005#	2006#
2007#	2008	2009#	2010	2011#	2012#	2013#	2014	2015#	2016	2017#	2018#	2019#	2020	2021#
2022	2023#	2024#	2025#	2026#	2027	2028#	2029	2030#	2031#	2032	2033#	2034	2036#	2039#
2040#	2041#	2042#	2043	2044#	2045	2046#	2047#	2048#	2049	2050#	2051	2052#	2053#	2054#
2055	2056#	2057	2058#	2059#	2060#	2061#	2062#	2063	2064#	2065	2066#	2067#	2068	2069#
2070	2072#	2075#	2076#	2077#	2078#	2079	2080#	2081	2082#	2083#	2084#	2085	2086#	2087
2088#	2089#	2090#	2091	2092#	2093	2094#	2095#	2096#	2097#	2098#	2099	2100#	2101	2102#

L 15

2103#	2104	2105#	2106	2108#	2111#	2112#	2113#	2114#	2115	2116#	2117	2118#	2119#	2120#
2121	2122#	2123	2124#	2125#	2126#	2127	2128#	2129	2130#	2131#	2132#	2133#	2134	2135#
2136	2137#	2138#	2139	2140#	2141	2143#	2146#	2147#	2148#	2149#	2150	2151#	2152	2153#
2154#	2155#	2156	2157#	2158	2159#	2160#	2161#	2162	2163#	2164	2165#	2166#	2167#	2168#
2169	2170#	2171	2172#	2173#	2174	2175#	2176	2178#	2181#	2182#	2183#	2184#	2185	2186#
2187	2188#	2189#	2190#	2191	2192#	2193	2194#	2195#	2196#	2197	2198#	2199	2200#	2201#
2202#	2203#	2204	2205#	2206	2207#	2208#	2209	2210#	2211	2213#	2216#	2217#	2218#	2219#
2220	2221#	2222	2223#	2224#	2225#	2226	2227#	2228	2229#	2230#	2231#	2232	2233#	2234
2235#	2236#	2237#	2238#	2239	2240#	2241	2242#	2243#	2244	2245#	2246	2248#	2251#	2252#
2253#	2254#	2255	2256#	2257	2258#	2259#	2260#	2261	2262#	2263	2264#	2265#	2266#	2267
2268#	2269	2270#	2271#	2272#	2273#	2274	2275#	2276	2277#	2278#	2279	2280#	2281	2283#
2286#	2287#	2288#	2289#	2290	2291#	2292	2293#	2294#	2295#	2296	2297#	2298	2299#	2300#
2301#	2302	2303#	2304	2305#	2306#	2307#	2308#	2309#	2310	2311#	2312	2313#	2314#	2315
2316#	2317	2319#	2322#	2323#	2324#	2325#	2326	2327#	2328	2329#	2330#	2331#	2332	2333#
2334	2335#	2336#	2337#	2338	2339#	2340	2341#	2342#	2343#	2344#	2345	2346#	2347	2348#
2349#	2350	2351#	2352	2354#	2359#	2360#	2361#	2362#	2363	2364#	2365	2366#	2367#	2368#
2369	2370#	2371	2372#	2373#	2374#	2375	2376#	2377	2378#	2379#	2380	2381#	2382	2384#
2387#	2388#	2389#	2390#	2391	2392#	2393	2394#	2395#	2396#	2397	2398#	2399	2400#	2401#
2402#	2403	2404#	2405	2406#	2407#	2408#	2409#	2410	2411#	2412	2413#	2414#	2415	2416#
2417	2419#	2422#	2423#	2424#	2425#	2426	2427#	2428	2429#	2430#	2431#	2432	2433#	2434
2435#	2436#	2437#	2438	2439#	2440	2441#	2442#	2443#	2444#	2445	2446#	2447	2448#	2449#
2450	2451#	2452	2454#	2459#	2460#	2461#	2462#	2463	2464#	2465	2467#	2468#	2469	2470#
2471	2474#	2479#	2480#	2481#	2482	2483#	2484	2486#	2487#	2488	2489#	2490	2493#	2497#
2498#	2499	2500#	2501	2503#	2504#	2505	2506#	2507	2510#	2514#	2515#	2516	2517#	2518
2520#	2521#	2522	2523#	2524	2527#	2532#	2533#	2534#	2535#	2536#	2537#	2538	2540#	2541#
2542	2543#	2544	2547#	2551#	2552#	2553#	2554#	2555	2556#	2557	2559#	2560#	2561	2562#
2563	2566#	2570#	2571#	2572#	2573#	2574	2575#	2576	2578#	2579#	2580	2581#	2582	2585#
2589#	2590#	2591#	2592#	2593	2594#	2595	2597#	2598#	2599	2600#	2601	2604#	2619#	2620#
2625#	2638#	2665#	2666#	2668#	2671#	2672#	2674#	2676#	2677#	2679#	2683#	2684#	2686#	2698#
2699#	2700#	2702#	2786#	2807#	2824#	2827#	2845#	2869#	2870#	2878#	2879#	2880#	2881#	2886#
2887#	2891#	2920#	2921#	2922#	2923#	2926#	2927#	2938#	2939#	2940#	2941#	2957#	2974#	2983#
2984#	2985#	2986#	2988#	2989#	2992#	3013#	3029#	3039#	3040#	3041#	3042#	3044#	3045#	3048#
3070#	3086#	3099#	3100#	3101#	3102#	3104#	3105#	3108#	3110#	3122#	3123#	3124#	3125#	3127#
3128#	3131#	3150#	3185#	3186#	3187#	3188#	3190#	3191#	3213#	3249#	3250#	3251#	3252#	3254#
3255#	3277#	3313#	3314#	3315#	3316#	3318#	3319#	3342#	3343#	3344#	3345#	3347#	3348#	3366#
3404#	3405#	3406#	3407#	3426#	3427#	3428#	3429#	3434#	3435#	3466#	3467#	3468#	3469#	3484#
3514#	3515#	3516#	3517#	3519#	3520#	3535#	3536#	3537#	3538#	3551#	3552#	3553#	3554#	3569#
3590#	3603#	3604#	3605#	3606#	3608#	3609#	3612#	3614#	3627#	3628#	3629#	3630#	3632#	3633#
3636#	3638#	3649#	3650#	3651#	3652#	3654#	3655#	3658#	3672#	3693#	3703#	3704#	3705#	3706#
3708#	3709#	3712#	3714#	3724#	3725#	3726#	3727#	3729#	3730#	3733#	3735#	3745#	3746#	3747#
3748#	3750#	3751#	3754#	3768#	3789#	3800#	3801#	3802#	3803#	3806#	3807#	3810#	3812#	3823#
3824#	3825#	3826#	3829#	3830#	3833#	3835#	3846#	3847#	3848#	3849#	3851#	3852#	3855#	3869#
3890#	3901#	3902#	3903#	3904#	3906#	3907#	3910#	3912#	3923#	3924#	3925#	3926#	3928#	3929#
3932#	3934#	3945#	3946#	3947#	3948#	3950#	3951#	3954#	3968#	3990#	4001#	4002#	4003#	4004#
4006#	4007#	4010#	4012#	4023#	4024#	4025#	4026#	4028#	4029#	4032#	4034#	4045#	4046#	4047#

4048#	4050#	4051#	4054#	4068#	4090#	4101#	4102#	4103#	4104#	4106#	4107#	4110#	4112#	4123#	
4124#	4125#	4126#	4128#	4129#	4132#	4134#	4145#	4146#	4147#	4148#	4150#	4151#	4154#	4168#	
4189#	4200#	4201#	4202#	4203#	4205#	4206#	4209#	4211#	4222#	4223#	4224#	4225#	4227#	4228#	
4231#	4233#	4244#	4245#	4246#	4247#	4249#	4250#	4253#	4267#	4288#	4299#	4300#	4301#	4302#	
4304#	4305#	4308#	4310#	4321#	4322#	4323#	4324#	4326#	4327#	4330#	4332#	4343#	4344#	4345#	
4346#	4348#	4349#	4352#	4366#	4386#	4397#	4398#	4399#	4400#	4402#	4403#	4406#	4408#	4421#	
4422#	4423#	4424#	4426#	4427#	4430#	4432#	4443#	4444#	4445#	4446#	4448#	4449#	4452#	4466#	
4486#	4497#	4498#	4499#	4500#	4502#	4503#	4506#	4508#	4521#	4522#	4523#	4524#	4526#	4527#	
4530#	4532#	4543#	4544#	4545#	4546#	4548#	4549#	4552#	4566#	4586#	4597#	4598#	4599#	4600#	
4602#	4603#	4606#	4608#	4621#	4622#	4623#	4624#	4626#	4627#	4630#	4632#	4643#	4644#	4645#	
4646#	4648#	4649#	4652#	4666#	4686#	4697#	4698#	4699#	4700#	4702#	4703#	4706#	4708#	4721#	
4722#	4723#	4724#	4726#	4727#	4730#	4732#	4743#	4744#	4745#	4746#	4748#	4749#	4752#	4766#	
4786#	4797#	4798#	4799#	4800#	4802#	4803#	4806#	4808#	4821#	4822#	4823#	4824#	4826#	4827#	
4830#	4832#	4843#	4844#	4845#	4846#	4848#	4849#	4852#	4866#	4886#	4897#	4898#	4899#	4900#	
4902#	4903#	4906#	4908#	4921#	4922#	4923#	4924#	4926#	4927#	4930#	4932#	4943#	4944#	4945#	
4946#	4948#	4949#	4952#	4966#	5016#	5017#	5018#	5019#	5025#	5026#	5044#	5084#	5085#	5086#	
5087#	5092#	5093#	5113#	5184#	5185#	5186#	5187#	5219#	5220#	5257#	5258#	5259#	5260#	5276#	
5300#	5301#	5302#	5303#	5306#	5307#	5318#	5319#	5320#	5321#	5324#	5325#	5334#	5335#	5336#	
5337#	5352#	5366#	5367#	5379#	5380#	5381#	5382#	5385#	5386#	5396#	5397#	5398#	5399#	5402#	
5403#	5412#	5413#	5414#	5415#	5430#	5455#	5456#	5457#	5458#	5471#	5472#	5473#	5474#	5485#	
5486#	5487#	5488#	5504#	5538#	5539#	5540#	5541#	5560#	5561#	5562#	5563#	5571#	5572#	5573#	
5574#	5579#	5580#	5588#	5589#	5590#	5591#	5601#	5602#	5603#	5604#	5613#	5721#	5722#	5723#	
5724#	5747#	5748#	5749#	5750#	5766#	5767#	5768#	5769#	5785#	5786#	5787#	5788#	5804#	5805#	
5828#	5869#	5870#	5871#	5872#	5888#	5928#	5929#	5930#	5931#	5935#	5936#	5952#	5953#	5961#	
6023#	6024#	6025#	6026#	6029#	6030#	6038#	6039#	6040#	6041#	6045#	6046#	6053#	6054#	6055#	
6056#	6059#	6060#	6067#	6068#	6069#	6070#	6073#	6074#	6083#	6084#	6085#	6086#	6089#	6090#	
6095#	6096#	6097#	6098#	6100#	6101#	6107#	6108#	6109#	6110#	6112#	6113#	6119#	6120#	6121#	
6122#	6137#	6138#	6159#	6204#	6205#	6223#	6224#	6225#	6226#	6231#	6253#	6254#	6255#	6256#	
6258#	6259#	6267#	6268#	6269#	6270#	6287#	6327#	6328#	6329#	6330#	6334#	6335#	6356#	6390#	
6391#	6392#	6393#	6410#	6446#	6447#	6448#	6449#	6466#	6508#	6509#	6510#	6511#	6528#	6548#	
6549#	6567#	6569#	6570#	6571#	6572#	6575#	6593#	6594#	6595#	6596#	6599#	6613#	6614#	6615#	
6616#	6620#	6623#	6633#	6634#	6635#	6636#	6655#	6670#	6675#	6676	6677	6678	6686#	6687	
6688	6689	6690	6693#	6775#	6780#	6791#	6792#	6793#							
MSGNLS	2991#	3047#	3107#	3130#	3611#	3635#	3657#	3711#	3732#	3753#	3809#	3832#	3854#	3909#	3931#
	3953#	4009#	4031#	4053#	4109#	4131#	4153#	4208#	4230#	4252#	4307#	4329#	4351#	4405#	4429#
	4451#	4505#	4529#	4551#	4605#	4629#	4651#	4705#	4729#	4751#	4805#	4829#	4851#	4905#	4929#
	4951#														
MSGNTA	889#	904#	1893#	1929#	1965#	2000#	2035#	2071#	2107#	2142#	2177#	2212#	2247#	2282#	2318#
	2353#	2383#	2418#	2453#	2473#	2492#	2509#	2526#	2546#	2565#	2584#	2603#	2624#	2637#	2785#
	2806#	2826#	2844#	2890#	2956#	3012#	3069#	3149#	3212#	3276#	3365#	3483#	3568#	3671#	3767#
	3868#	3967#	4067#	4167#	4266#	4365#	4465#	4565#	4665#	4765#	4865#	4965#	5043#	5112#	5275#
	5351#	5429#	5503#	5612#	5827#	5887#	5960#	6158#	6230#	6286#	6355#	6409#	6465#	6527#	6654#
	6693#	6694	6780#	6781											
MSGNTE	2861#	2902#	2966#	3022#	3080#	3159#	3222#	3287#	3376#	3495#	3584#	3687#	3783#	3884#	3983#
	4083#	4183#	4282#	4381#	4481#	4581#	4681#	4781#	4881#	4979#	5058#	5151#	5288#	5364#	5441#
	5519#	5626#	5840#	5899#	5972#	6172#	6243#	6299#	6371#	6425#	6481#	6542#			
MSHAPT	714#														
MSHNAP	714#	753													
MSINCR	688#	874#	898#	1860#	1866#	1872#	1878#	1886#	1891#	1894#	1896#	1902#	1908#	1914#	1922#
	1927#	1930#	1932#	1938#	1944#	1950#	1958#	1963#	1966#	1968#	1974#	1980#	1986#	1993#	1998#
	2001#	2003#	2009#	2015#	2021#	2028#	2033#	2036#	2038#	2044#	2050#	2056#	2064#	2069#	2072#
	2074#	2080#	2086#	2092#	2100#	2105#	2108#	2110#	2116#	2122#	2128#	2135#	2140#	2143#	2145#
	2151#	2157#	2163#	2170#	2175#	2178#	2180#	2186#	2192#	2198#	2205#	2210#	2213#	2215#	2221#
	2227#	2233#	2240#	2245#	2248#	2250#	2256#	2262#	2268#	2275#	2280#	2283#	2285#	2291#	2297#
	2303#	2311#	2316#	2319#	2321#	2327#	2333#	2339#	2346#	2351#	2354#	2358#	2364#	2370#	2376#
	2381#	2384#	2386#	2392#	2398#	2404#	2411#	2416#	2419#	2421#	2427#	2433#	2439#	2446#	2451#

2454#	2457#	2464#	2470#	2474#	2477#	2483#	2489#	2493#	2495#	2500#	2506#	2510#	2512#	2517#	
2523#	2527#	2530#	2537#	2543#	2547#	2549#	2556#	2562#	2566#	2568#	2575#	2581#	2585#	2587#	
2594#	2600#	2604#	2615#	2625#	2628#	2635#	2638#	2649#	2666#	2672#	2677#	2684#	2699#	2786#	
2802#	2807#	2821#	2824#	2827#	2842#	2845#	2861#	2862#	2869#	2878#	2886#	2891#	2902#	2903#	
2920#	2926#	2938#	2957#	2966#	2967#	2974#	2983#	2988#	2992#	3013#	3022#	3023#	3029#	3039#	
3044#	3048#	3070#	3080#	3081#	3086#	3099#	3104#	3108#	3110#	3122#	3127#	3131#	3150#	3159#	
3160#	3185#	3190#	3213#	3222#	3223#	3249#	3254#	3277#	3287#	3288#	3313#	3318#	3342#	3347#	
3366#	3376#	3377#	3404#	3426#	3434#	3466#	3484#	3495#	3496#	3514#	3519#	3535#	3551#	3569#	
3584#	3585#	3590#	3603#	3608#	3612#	3614#	3627#	3632#	3636#	3638#	3649#	3654#	3658#	3672#	
3687#	3688#	3693#	3703#	3708#	3712#	3714#	3724#	3729#	3733#	3735#	3745#	3750#	3754#	3768#	
3783#	3784#	3789#	3800#	3806#	3810#	3812#	3823#	3829#	3833#	3835#	3846#	3851#	3855#	3869#	
3884#	3885#	3890#	3901#	3906#	3910#	3912#	3923#	3928#	3932#	3934#	3945#	3950#	3954#	3968#	
3983#	3984#	3990#	4001#	4006#	4010#	4012#	4023#	4028#	4032#	4034#	4045#	4050#	4054#	4068#	
4083#	4084#	4090#	4101#	4106#	4110#	4112#	4123#	4128#	4132#	4134#	4145#	4150#	4154#	4168#	
4183#	4184#	4189#	4200#	4205#	4209#	4211#	4222#	4227#	4231#	4233#	4244#	4249#	4253#	4267#	
4282#	4283#	4288#	4299#	4304#	4308#	4310#	4321#	4326#	4330#	4332#	4343#	4348#	4352#	4366#	
4381#	4382#	4386#	4397#	4402#	4406#	4408#	4421#	4426#	4430#	4432#	4443#	4448#	4452#	4466#	
4481#	4482#	4486#	4497#	4502#	4506#	4508#	4521#	4526#	4530#	4532#	4543#	4548#	4552#	4566#	
4581#	4582#	4586#	4597#	4602#	4606#	4608#	4621#	4626#	4630#	4632#	4643#	4648#	4652#	4666#	
4681#	4682#	4686#	4697#	4702#	4706#	4708#	4721#	4726#	4730#	4732#	4743#	4748#	4752#	4766#	
4781#	4782#	4786#	4797#	4802#	4806#	4808#	4821#	4826#	4830#	4832#	4843#	4848#	4852#	4866#	
4881#	4882#	4886#	4897#	4902#	4906#	4908#	4921#	4926#	4930#	4932#	4943#	4948#	4952#	4966#	
4979#	4980#	5016#	5025#	5044#	5058#	5059#	5084#	5092#	5113#	5151#	5152#	5184#	5219#	5257#	
5276#	5288#	5289#	5300#	5306#	5318#	5324#	5334#	5352#	5364#	5365#	5366#	5379#	5385#	5396#	
5402#	5412#	5430#	5441#	5442#	5455#	5471#	5485#	5504#	5519#	5520#	5538#	5560#	5571#	5579#	
5588#	5601#	5613#	5626#	5627#	5627#	5747#	5766#	5785#	5804#	5828#	5840#	5841#	5869#	5888#	
5899#	5900#	5928#	5935#	5952#	5961#	5972#	5973#	6023#	6029#	6038#	6045#	6053#	6059#	6067#	
6073#	6083#	6089#	6095#	6100#	6107#	6112#	6119#	6137#	6159#	6172#	6173#	6204#	6223#	6231#	
6243#	6244#	6253#	6258#	6267#	6287#	6299#	6300#	6327#	6334#	6356#	6371#	6372#	6390#	6410#	
6425#	6426#	6446#	6466#	6481#	6482#	6508#	6528#	6542#	6543#	6549#	6567#	6569#	6575#	6593#	
6599#	6613#	6620#	6623#	6633#	6655#	6670#	6775#								
MSLDRO	2665#	2671#	2676#	2683#	2698#	6548#									
MSMCHI	682#														
MSMCLO	682#														
MSPOP	889#	904#	1893#	1929#	1965#	2000#	2035#	2071#	2107#	2142#	2177#	2212#	2247#	2282#	2318#
	2353#	2383#	2418#	2453#	2473#	2492#	2509#	2526#	2546#	2565#	2584#	2603#	2624#	2633#	2637#
	2785#	2806#	2826#	2844#	2890#	2956#	2991#	3012#	3047#	3069#	3107#	3130#	3149#	3212#	3276#
	3365#	3483#	3568#	3611#	3635#	3657#	3671#	3711#	3732#	3753#	3767#	3809#	3832#	3854#	3868#
	3909#	3931#	3953#	3967#	4009#	4031#	4053#	4067#	4109#	4131#	4153#	4167#	4208#	4230#	4252#
	4266#	4307#	4329#	4351#	4365#	4405#	4429#	4451#	4465#	4505#	4529#	4551#	4565#	4605#	4629#
	4651#	4665#	4705#	4729#	4751#	4765#	4805#	4829#	4851#	4865#	4905#	4929#	4951#	4965#	5043#
	5112#	5275#	5351#	5429#	5503#	5612#	5827#	5887#	5960#	6158#	6230#	6286#	6355#	6409#	6465#
	6527#	6654#	6693#	6780#											
MSPRIN	1861#	1868#	1874#	1880#	1888#	1897#	1904#	1910#	1916#	1924#	1933#	1940#	1946#	1952#	1960#
	1969#	1976#	1982#	1988#	1995#	2004#	2011#	2017#	2023#	2030#	2039#	2046#	2052#	2058#	2066#
	2075#	2082#	2088#	2094#	2102#	2111#	2118#	2124#	2130#	2137#	2146#	2153#	2159#	2165#	2172#
	2181#	2188#	2194#	2200#	2207#	2216#	2223#	2229#	2235#	2242#	2251#	2258#	2264#	2270#	2277#
	2286#	2293#	2299#	2305#	2313#	2322#	2329#	2335#	2341#	2348#	2359#	2366#	2372#	2378#	2387#
	2394#	2400#	2406#	2413#	2422#	2429#	2435#	2441#	2448#	2459#	2467#	2479#	2486#	2497#	2503#
	2514#	2520#	2532#	2540#	2551#	2559#	2570#	2578#	2589#	2597#					
MSPUSH	688#	874#	898#	1860#	1896#	1932#	1968#	2003#	2038#	2074#	2110#	2145#	2180#	2215#	2250#
	2285#	2321#	2358#	2386#	2421#	2457#	2477#	2495#	2512#	2530#	2549#	2568#	2587#	2615#	2628#
	2635#	2649#	2802#	2821#	2842#	2861#	2862	2902#	2903	2966#	2967	2974#	3022#	3023	3029#
	3080#	3081	3086#	3110#	3159#	3160	3222#	3223	3287#	3288	3376#	3377	3495#	3496	3584#
	3585	3590#	3614#	3638#	3687#	3688	3693#	3714#	3735#	3783#	3784	3789#	3812#	3835#	3884#
	3885	3890#	3912#	3934#	3983#	3984	3990#	4012#	4034#	4083#	4084	4090#	4112#	4134#	4183#

	4184	4189#	4211#	4233#	4282#	4283	4288#	4310#	4332#	4381#	4382	4386#	4408#	4432#	4481#
	4482	4486#	4508#	4532#	4581#	4582	4586#	4608#	4632#	4681#	4682	4686#	4708#	4732#	4781#
	4782	4786#	4808#	4832#	4881#	4882	4886#	4908#	4932#	4979#	4980	5058#	5059	5151#	5152
	5288#	5289	5364#	5365	5441#	5442	5519#	5520	5626#	5627	5840#	5841	5899#	5900	5972#
	5973	6172#	6173	6243#	6244	6299#	6300	6371#	6372	6425#	6426	6481#	6482	6542#	6543
MSPUT	6670#	6775#													
	1861#	1868#	1874#	1880#	1888#	1897#	1904#	1910#	1916#	1924#	1933#	1940#	1946#	1952#	1960#
	1969#	1976#	1982#	1988#	1995#	2004#	2011#	2017#	2023#	2030#	2039#	2046#	2052#	2058#	2066#
	2075#	2082#	2088#	2094#	2102#	2111#	2118#	2124#	2130#	2137#	2146#	2153#	2159#	2165#	2172#
	2181#	2188#	2194#	2200#	2207#	2216#	2223#	2229#	2235#	2242#	2251#	2258#	2264#	2270#	2277#
	2286#	2293#	2299#	2305#	2313#	2322#	2329#	2335#	2341#	2348#	2359#	2366#	2372#	2378#	2387#
	2394#	2400#	2406#	2413#	2422#	2429#	2435#	2441#	2448#	2459#	2467#	2479#	2486#	2497#	2503#
MSPUT1	2514#	2520#	2532#	2540#	2551#	2559#	2570#	2578#	2589#	2597#					
	1861#	1862	1863	1864	1868#	1869	1870	1874#	1875	1876	1880#	1881	1882	1883	1884
	1888#	1889	1897#	1898	1899	1900	1904#	1905	1906	1910#	1911	1912	1916#	1917	1918
	1919	1920	1924#	1925	1933#	1934	1935	1936	1940#	1941	1942	1946#	1947	1948	1952#
	1953	1954	1955	1956	1960#	1961	1969#	1970	1971	1972	1976#	1977	1978	1982#	1983
	1984	1988#	1989	1990	1991	1995#	1996	2004#	2005	2006	2007	2011#	2012	2013	2017#
	2018	2019	2023#	2024	2025	2026	2030#	2031	2039#	2040	2041	2042	2046#	2047	2048
	2052#	2053	2054	2058#	2059	2060	2061	2062	2066#	2067	2075#	2076	2077	2078	2082#
	2083	2084	2088#	2089	2090	2094#	2095	2096	2097	2098	2102#	2103	2111#	2112	2113
	2114	2118#	2119	2120	2124#	2125	2126	2130#	2131	2132	2133	2137#	2138	2146#	2147
	2148	2149	2153#	2154	2155	2159#	2160	2161	2165#	2166	2167	2168	2172#	2173	2181#
	2182	2183	2184	2188#	2189	2190	2194#	2195	2196	2200#	2201	2202	2203	2207#	2208
	2216#	2217	2218	2219	2223#	2224	2225	2229#	2230	2231	2235#	2236	2237	2238	2242#
	2243	2251#	2252	2253	2254	2258#	2259	2260	2264#	2265	2266	2270#	2271	2272	2273
	2277#	2278	2286#	2287	2288	2289	2293#	2294	2295	2299#	2300	2301	2305#	2306	2307
	2308	2309	2313#	2314	2322#	2323	2324	2325	2329#	2330	2331	2335#	2336	2337	2341#
	2342	2343	2344	2348#	2349	2359#	2360	2361	2362	2366#	2367	2368	2372#	2373	2374
	2378#	2379	2387#	2388	2389	2390	2394#	2395	2396	2400#	2401	2402	2406#	2407	2408
	2409	2413#	2414	2422#	2423	2424	2425	2429#	2430	2431	2435#	2436	2437	2441#	2442
	2443	2444	2448#	2449	2459#	2460	2461	2462	2467#	2468	2479#	2480	2481	2486#	2487
	2497#	2498	2503#	2504	2514#	2515	2520#	2521	2532#	2533	2534	2535	2540#	2541	2551#
	2552	2553	2554	2559#	2560	2570#	2571	2572	2573	2578#	2579	2589#	2590	2591	2592
	2597#	2598													
MSRADI	6675#	6686#													
MSRNRO	2698#	2700													
MSSETS	688#	874#	898#	1860#	1896#	1932#	1968#	2003#	2038#	2074#	2110#	2145#	2180#	2215#	2250#
	2285#	2321#	2358#	2386#	2421#	2457#	2477#	2495#	2512#	2530#	2549#	2568#	2587#	2615#	2628#
	2635#	2649#	2802#	2821#	2842#	2862#	2903#	2967#	2974#	3023#	3029#	3081#	3086#	3110#	3160#
	3223#	3288#	3377#	3496#	3585#	3590#	3614#	3638#	3688#	3693#	3714#	3735#	3784#	3789#	3812#
	3835#	3885#	3890#	3912#	3934#	3984#	3990#	4012#	4034#	4084#	4090#	4112#	4134#	4184#	4189#
	4211#	4233#	4283#	4288#	4310#	4332#	4382#	4386#	4408#	4432#	4482#	4486#	4508#	4532#	4582#
	4586#	4608#	4632#	4682#	4686#	4708#	4732#	4782#	4786#	4808#	4832#	4882#	4886#	4908#	4932#
	4980#	5059#	5152#	5289#	5365#	5442#	5520#	5627#	5841#	5900#	5973#	6173#	6244#	6300#	6372#
	6426#	6482#	6543#	6670#	6775#										
MSSVC	1861#	1866	1868#	1872	1874#	1878	1880#	1886	1888#	1891	1893#	1894	1897#	1902	1904#
	1908	1910#	1914	1916#	1922	1924#	1927	1929#	1930	1933#	1938	1940#	1944	1946#	1950
	1952#	1958	1960#	1963	1965#	1966	1969#	1974	1976#	1980	1982#	1986	1988#	1993	1995#
	1998	2000#	2001	2004#	2009	2011#	2015	2017#	2021	2023#	2028	2030#	2033	2035#	2036
	2039#	2044	2046#	2050	2052#	2056	2058#	2064	2066#	2069	2071#	2072	2075#	2080	2082#
	2086	2088#	2092	2094#	2100	2102#	2105	2107#	2108	2111#	2116	2118#	2122	2124#	2128
	2130#	2135	2137#	2140	2142#	2143	2146#	2151	2153#	2157	2159#	2163	2165#	2170	2172#
	2175	2177#	2178	2181#	2186	2188#	2192	2194#	2198	2200#	2205	2207#	2210	2212#	2213
	2216#	2221	2223#	2227	2229#	2233	2235#	2240	2242#	2245	2247#	2248	2251#	2256	2258#
	2262	2264#	2268	2270#	2275	2277#	2280	2282#	2283	2286#	2291	2293#	2297	2299#	2303

2305#	2311	2313#	2316	2318#	2319	2322#	2327	2329#	2333	2335#	2339	2341#	2346	2348#	
2351	2353#	2354	2359#	2364	2366#	2370	2372#	2376	2378#	2381	2383#	2384	2387#	2392	
2394#	2398	2400#	2404	2406#	2411	2413#	2416	2418#	2419	2422#	2427	2429#	2433	2435#	
2439	2441#	2446	2448#	2451	2453#	2454	2459#	2464	2467#	2470	2473#	2474	2479#	2483	
2486#	2489	2492#	2493	2497#	2500	2503#	2506	2509#	2510	2514#	2517	2520#	2523	2526#	
2527	2532#	2537	2540#	2543	2546#	2547	2551#	2556	2559#	2562	2565#	2566	2570#	2575	
2578#	2581	2584#	2585	2589#	2594	2597#	2600	2603#	2604	2619#	2624#	2625	2637#	2638	
2665#	2666	2671#	2672	2676#	2677	2683#	2684	2698#	2699	2785#	2786	2806#	2807	2824#	
2826#	2827	2844#	2845	2869#	2878	2886#	2890#	2891	2920	2926#	2938	2956#	2957	2974#	
2983	2988#	2991#	2992	3012#	3013	3029#	3039	3044#	3047#	3048	3069#	3070	3086#	3099	
3104#	3107#	3108	3110#	3122	3127#	3130#	3131	3149#	3150	3185	3190#	3212#	3213	3249	
3254#	3276#	3277	3313	3318#	3342	3347#	3365#	3366	3404	3426	3434#	3466	3483#	3484	
3514	3519#	3535	3551	3568#	3569	3590#	3603	3608#	3611#	3612	3614#	3627	3632#	3635#	
3636	3638#	3649	3654#	3657#	3658	3671#	3672	3693#	3703	3708#	3711#	3712	3714#	3724	
3729#	3732#	3733	3735#	3745	3750#	3753#	3754	3767#	3768	3789#	3800	3806#	3809#	3810	
3812#	3823	3829#	3832#	3833	3835#	3846	3851#	3854#	3855	3868#	3869	3890#	3901	3906#	
3909#	3910	3912#	3923	3928#	3931#	3932	3934#	3945	3950#	3953#	3954	3967#	3968	3990#	
4001	4006#	4009#	4010	4012#	4023	4028#	4031#	4032	4034#	4045	4050#	4053#	4054	4067#	
4068	4090#	4101	4106#	4109#	4110	4112#	4123	4128#	4131#	4132	4134#	4145	4150#	4153#	
4154	4167#	4168	4189#	4200	4205#	4208#	4209	4211#	4222	4227#	4230#	4231	4233#	4244	
4249#	4252#	4253	4266#	4267	4288#	4299	4304#	4307#	4308	4310#	4321	4326#	4329#	4330	
4332#	4343	4348#	4351#	4352	4365#	4366	4386#	4397	4402#	4405#	4406	4408#	4421	4426#	
4429#	4430	4432#	4443	4448#	4451#	4452	4465#	4466	4486#	4497	4502#	4505#	4506	4508#	
4521	4526#	4529#	4530	4532#	4543	4548#	4551#	4552	4565#	4566	4586#	4597	4602#	4605#	
4606	4608#	4621	4626#	4629#	4630	4632#	4643	4648#	4651#	4652	4665#	4666	4686#	4697	
4702#	4705#	4706	4708#	4721	4726#	4729#	4730	4732#	4743	4748#	4751#	4752	4765#	4766	
4786#	4797	4802#	4805#	4806	4808#	4821	4826#	4829#	4830	4832#	4843	4848#	4851#	4852	
4865#	4866	4886#	4897	4902#	4905#	4906	4908#	4921	4926#	4929#	4930	4932#	4943	4948#	
4951#	4952	4965#	4966	5016	5025#	5043#	5044	5084	5092#	5112#	5113	5184	5219#	5257	
5275#	5276	5300	5306#	5318	5324#	5334	5351#	5352	5366#	5379	5385#	5396	5402#	5412	
5429#	5436	5455	5471	5485	5503#	5504	5538	5560	5571	5579#	5588	5601	5612#	5613	
5721	5747	5766	5785	5804#	5827#	5828	5869	5887#	5888	5928	5935#	5952#	5960#	5961	
6023	6029#	6038	6045#	6053	6059#	6067	6073#	6083	6089#	6095	6100#	6107	6112#	6119	
6137#	6158#	6159	6204#	6223	6230#	6231	6253	6258#	6267	6286#	6287	6327	6334#	6355#	
6356	6390	6409#	6410	6446	6465#	6466	6508	6527#	6528	6548#	6549	6567#	6569	6575#	
6593	6599#	6613	6620#	6623#	6633	6654#	6655								
MSTLAB	1866#	1872#	1878#	1886#	1891#	1894#	1902#	1908#	1914#	1922#	1927#	1930#	1938#	1944#	1950#
	1958#	1963#	1966#	1974#	1980#	1986#	1993#	1998#	2001#	2009#	2015#	2021#	2028#	2033#	2036#
	2044#	2050#	2056#	2064#	2069#	2072#	2080#	2086#	2092#	2100#	2105#	2108#	2116#	2122#	2128#
	2135#	2140#	2143#	2151#	2157#	2163#	2170#	2175#	2178#	2186#	2192#	2198#	2205#	2210#	2213#
	2221#	2227#	2233#	2240#	2245#	2248#	2256#	2262#	2268#	2275#	2280#	2283#	2291#	2297#	2303#
	2311#	2316#	2319#	2327#	2333#	2339#	2346#	2351#	2354#	2364#	2370#	2376#	2381#	2384#	2392#
	2398#	2404#	2411#	2416#	2419#	2427#	2433#	2439#	2446#	2451#	2454#	2464#	2470#	2474#	2483#
	2489#	2493#	2500#	2506#	2510#	2517#	2523#	2527#	2537#	2543#	2547#	2556#	2562#	2566#	2575#
	2581#	2585#	2594#	2600#	2604#	2625#	2638#	2666#	2672#	2677#	2684#	2699#	2786#	2807#	2824#
	2827#	2845#	2869#	2878#	2886#	2891#	2920#	2926#	2938#	2957#	2974#	2983#	2988#	2992#	3013#
	3029#	3039#	3044#	3048#	3070#	3086#	3099#	3104#	3108#	3110#	3122#	3127#	3131#	3150#	3185#
	3190#	3213#	3249#	3254#	3277#	3313#	3318#	3342#	3347#	3366#	3404#	3426#	3434#	3466#	3484#
	3514#	3519#	3535#	3551#	3569#	3590#	3603#	3608#	3612#	3614#	3627#	3632#	3636#	3638#	3649#
	3654#	3658#	3672#	3693#	3703#	3708#	3712#	3714#	3724#	3729#	3733#	3735#	3745#	3750#	3754#
	3768#	3789#	3800#	3806#	3810#	3812#	3823#	3829#	3833#	3835#	3846#	3851#	3855#	3869#	3890#
	3901#	3906#	3910#	3912#	3923#	3928#	3932#	3934#	3945#	3950#	3954#	3968#	3990#	4001#	4006#
	4010#	4012#	4023#	4028#	4032#	4034#	4045#	4050#	4054#	4068#	4090#	4101#	4106#	4110#	4112#
	4123#	4128#	4132#	4134#	4145#	4150#	4154#	4168#	4189#	4200#	4205#	4209#	4211#	4222#	4227#
	4231#	4233#	4244#	4249#	4253#	4267#	4288#	4299#	4304#	4308#	4310#	4321#	4326#	4330#	4332#
	4343#	4348#	4352#	4366#	4386#	4397#	4402#	4406#	4408#	4421#	4426#	4430#	4432#	4443#	4448#

	4452#	4466#	4486#	4497#	4502#	4506#	4508#	4521#	4526#	4530#	4532#	4543#	4548#	4552#	4566#
	4586#	4597#	4602#	4606#	4608#	4621#	4626#	4630#	4632#	4643#	4648#	4652#	4666#	4686#	4697#
	4702#	4706#	4708#	4721#	4726#	4730#	4732#	4743#	4748#	4752#	4766#	4786#	4797#	4802#	4806#
	4808#	4821#	4826#	4830#	4832#	4843#	4848#	4852#	4866#	4886#	4897#	4902#	4906#	4908#	4921#
	4926#	4930#	4932#	4943#	4948#	4952#	4966#	5016#	5025#	5044#	5084#	5092#	5113#	5184#	5219#
	5257#	5276#	5300#	5306#	5318#	5324#	5334#	5352#	5366#	5379#	5385#	5396#	5402#	5412#	5430#
	5455#	5471#	5485#	5504#	5538#	5560#	5571#	5579#	5588#	5601#	5613#	5721#	5747#	5766#	5785#
	5804#	5828#	5869#	5888#	5928#	5935#	5952#	5961#	6023#	6029#	6038#	6045#	6053#	6059#	6067#
	6073#	6083#	6089#	6095#	6100#	6107#	6112#	6119#	6137#	6159#	6204#	6223#	6231#	6253#	6258#
	6267#	6287#	6327#	6334#	6356#	6390#	6410#	6446#	6466#	6508#	6528#	6549#	6567#	6569#	6575#
MSTSTL	6593#	6599#	6613#	6620#	6623#	6633#	6655#								
	1866#	1872#	1878#	1886#	1891#	1894#	1902#	1908#	1914#	1922#	1927#	1930#	1938#	1944#	1950#
	1958#	1963#	1966#	1974#	1980#	1986#	1993#	1998#	2001#	2009#	2015#	2021#	2028#	2033#	2036#
	2044#	2050#	2056#	2064#	2069#	2072#	2080#	2086#	2092#	2100#	2105#	2108#	2116#	2122#	2128#
	2135#	2140#	2143#	2151#	2157#	2163#	2170#	2175#	2178#	2186#	2192#	2198#	2205#	2210#	2213#
	2221#	2227#	2233#	2240#	2245#	2248#	2256#	2262#	2268#	2275#	2280#	2283#	2291#	2297#	2303#
	2311#	2316#	2319#	2327#	2333#	2339#	2346#	2351#	2354#	2364#	2370#	2376#	2381#	2384#	2392#
	2398#	2404#	2411#	2416#	2419#	2427#	2433#	2439#	2446#	2451#	2454#	2464#	2470#	2474#	2483#
	2489#	2493#	2500#	2506#	2510#	2517#	2523#	2527#	2537#	2543#	2547#	2556#	2562#	2566#	2575#
	2581#	2585#	2594#	2600#	2604#	2625#	2638#	2666#	2672#	2677#	2684#	2699#	2786#	2807#	2824#
	2827#	2845#	2869#	2878#	2886#	2891#	2920#	2926#	2938#	2957#	2974#	2983#	2988#	2992#	3013#
	3029#	3039#	3044#	3048#	3070#	3086#	3099#	3104#	3108#	3110#	3122#	3127#	3131#	3150#	3185#
	3190#	3213#	3249#	3254#	3277#	3313#	3318#	3342#	3347#	3366#	3404#	3426#	3434#	3466#	3484#
	3514#	3519#	3535#	3551#	3569#	3590#	3603#	3608#	3612#	3614#	3627#	3632#	3636#	3638#	3649#
	3654#	3658#	3672#	3693#	3703#	3708#	3712#	3714#	3724#	3729#	3733#	3735#	3745#	3750#	3754#
	3768#	3789#	3800#	3806#	3810#	3812#	3823#	3829#	3833#	3835#	3846#	3851#	3855#	3869#	3890#
	3901#	3906#	3910#	3912#	3923#	3928#	3932#	3934#	3945#	3950#	3954#	3968#	3990#	4001#	4006#
	4010#	4012#	4023#	4028#	4032#	4034#	4045#	4050#	4054#	4068#	4090#	4101#	4106#	4110#	4112#
	4123#	4128#	4132#	4134#	4145#	4150#	4154#	4168#	4189#	4200#	4205#	4209#	4211#	4222#	4227#
	4231#	4233#	4244#	4249#	4253#	4267#	4288#	4299#	4304#	4308#	4310#	4321#	4326#	4330#	4332#
	4343#	4348#	4352#	4366#	4386#	4397#	4402#	4406#	4408#	4421#	4426#	4430#	4432#	4443#	4448#
	4452#	4466#	4486#	4497#	4502#	4506#	4508#	4521#	4526#	4530#	4532#	4543#	4548#	4552#	4566#
	4586#	4597#	4602#	4606#	4608#	4621#	4626#	4630#	4632#	4643#	4648#	4652#	4666#	4686#	4697#
	4702#	4706#	4708#	4721#	4726#	4730#	4732#	4743#	4748#	4752#	4766#	4786#	4797#	4802#	4806#
	4808#	4821#	4826#	4830#	4832#	4843#	4848#	4852#	4866#	4886#	4897#	4902#	4906#	4908#	4921#
	4926#	4930#	4932#	4943#	4948#	4952#	4966#	5016#	5025#	5044#	5084#	5092#	5113#	5184#	5219#
	5257#	5276#	5300#	5306#	5318#	5324#	5334#	5352#	5366#	5379#	5385#	5396#	5402#	5412#	5430#
	5455#	5471#	5485#	5504#	5538#	5560#	5571#	5579#	5588#	5601#	5613#	5721#	5747#	5766#	5785#
	5804#	5828#	5869#	5888#	5928#	5935#	5952#	5961#	6023#	6029#	6038#	6045#	6053#	6059#	6067#
	6073#	6083#	6089#	6095#	6100#	6107#	6112#	6119#	6137#	6159#	6204#	6223#	6231#	6253#	6258#
	6267#	6287#	6327#	6334#	6356#	6390#	6410#	6446#	6466#	6508#	6528#	6549#	6567#	6569#	6575#
	6593#	6599#	6613#	6620#	6623#	6633#	6655#								
MSWORD	753#	762	814#	816	817	818	819	820	821	822	823	824	825	826	827
	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842
	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857
	2619#	2878#	2879	2880	2881	2920#	2921	2922	2923	2938#	2939	2940	2941	2983#	2984
	2985	2986	3039#	3040	3041	3042	3099#	3100	3101	3102	3122#	3123	3124	3125	3185#
	3186	3187	3188	3249#	3250	3251	3252	3313#	3314	3315	3316	3342#	3343	3344	3345
	3404#	3405	3406	3407	3426#	3427	3428	3429	3466#	3467	3468	3469	3514#	3515	3516
	3517	3535#	3536	3537	3538	3551#	3552	3553	3554	3603#	3604	3605	3606	3627#	3628
	3629	3630	3649#	3650	3651	3652	3703#	3704	3705	3706	3724#	3725	3726	3727	3745#
	3746	3747	3748	3800#	3801	3802	3803	3823#	3824	3825	3826	3846#	3847	3848	3849
	3901#	3902	3903	3904	3923#	3924	3925	3926	3945#	3946	3947	3948	4001#	4002	4003
	4004	4023#	4024	4025	4026	4045#	4046	4047	4048	4101#	4102	4103	4104	4123#	4124
	4125	4126	4145#	4146	4147	4148	4200#	4201	4202	4203	4222#	4223	4224	4225	4244#
	4245	4246	4247	4299#	4300	4301	4302	4321#	4322	4323	4324	4343#	4344	4345	4346

	4397#	4398	4399	4400	4421#	4422	4423	4424	4443#	4444	4445	4446	4497#	4498	4499
	4500	4521#	4522	4523	4524	4543#	4544	4545	4546	4597#	4598	4599	4600	4621#	4622
	4623	4624	4643#	4644	4645	4646	4697#	4698	4699	4700	4721#	4722	4723	4724	4743#
	4744	4745	4746	4797#	4798	4799	4800	4821#	4822	4823	4824	4843#	4844	4845	4846
	4897#	4898	4899	4900	4921#	4922	4923	4924	4943#	4944	4945	4946	5016#	5017	5018
	5019	5084#	5085	5086	5087	5184#	5185	5186	5187	5257#	5258	5259	5260	5300#	5301
	5302	5303	5318#	5319	5320	5321	5334#	5335	5336	5337	5366#	5379#	5380	5381	5382
	5396#	5397	5398	5399	5412#	5413	5414	5415	5455#	5456	5457	5458	5471#	5472	5473
	5474	5485#	5486	5487	5488	5538#	5539	5540	5541	5560#	5561	5562	5563	5571#	5572
	5573	5574	5579#	5588#	5589	5590	5591	5601#	5602	5603	5604	5721#	5722	5723	5724
	5747#	5748	5749	5750	5766#	5767	5768	5769	5785#	5786	5787	5788	5804#	5869#	5870
	5871	5872	5928#	5929	5930	5931	5952#	6023#	6024	6025	6026	6038#	6039	6040	6041
	6053#	6054	6055	6056	6067#	6068	6069	6070	6083#	6084	6085	6086	6095#	6096	6097
	6098	6107#	6108	6109	6110	6119#	6120	6121	6122	6137#	6204#	6223#	6224	6225	6226
	6253#	6254	6255	6256	6267#	6268	6269	6270	6327#	6328	6329	6330	6390#	6391	6392
	6393	6446#	6447	6448	6449	6508#	6509	6510	6511	6569#	6570	6571	6572	6593#	6594
	6595	6596	6613#	6614	6615	6616	6633#	6634	6635	6636	6675#	6686#	6792	6793	
POINTE	710														
POPSP2	1224#	5554													
PRINTB	1861	1868	1874	1880	1888	1897	1904	1910	1916	1924	1933	1940	1946	1952	1960
	1969	1976	1982	1988	1995	2004	2011	2017	2023	2030	2039	2046	2052	2058	2066
	2075	2082	2088	2094	2102	2111	2118	2124	2130	2137	2146	2153	2159	2165	2172
	2181	2188	2194	2200	2207	2216	2223	2229	2235	2242	2251	2258	2264	2270	2277
	2286	2293	2299	2305	2313	2322	2329	2335	2341	2348	2359	2366	2372	2378	2387
	2394	2400	2406	2413	2422	2429	2435	2441	2448	2466	2485	2502	2519	2539	2558
	2577	2596													
PRINTF	2458	2478	2496	2513	2531	2550	2569	2588							
READEF	2664	2670	2675	2682											
ROMCLK	1281#	1323	1471	1473	1475	1483	1490	1497	1504	1511	1513	1515	1524	1535	1537
	2909	2911	2913	2929	2931	3169	3171	3174	3176	3178	3233	3235	3238	3240	3242
	3297	3299	3302	3304	3306	3329	3331	3333	3335	3383	3386	3387	3393	3396	3411
	3414	3452	3454	3500	3503	3522	3527	3539	3541	3543	4985	4987	4991	5006	5008
	5067	5069	5076	5164	5166	5172	5174	5242	5244	5292	5309	5327	5371	5388	5405
	5447	5476	5633	5646	5649	5652	5665	5668	5671	5677	5679	5682	5685	5688	5696
	5698	5713	5741	5757	5760	5774	5776	5779	5843	5845	5847	5849	5853	5856	6316
	6319	6380	6434	6440	6487	6490	6501	6551	6554	6556	6559	6581	6584	6603	6605
	6625														
SETPRI	6547														
SKIP04	1296#														
SKIP06	1288#	3092													
SKIP07	1292#	1548													
SROMCL	1285#	1477	1517	1526	3397	3408	3448	3450	3594	3596	3618	3620	3640	3642	3694
	3696	3715	3717	3736	3738	3791	3793	3814	3816	3837	3839	3892	3894	3914	3916
	3936	3938	3992	3994	4014	4016	4036	4038	4092	4094	4114	4116	4136	4138	4191
	4193	4213	4215	4235	4237	4290	4292	4312	4314	4334	4336	4388	4390	4412	4414
	4434	4436	4488	4490	4512	4514	4534	4536	4588	4590	4612	4614	4634	4636	4688
	4690	4712	4714	4734	4736	4788	4790	4812	4814	4834	4836	4888	4890	4912	4914
	4934	4936	5071	5168	5176	5238	5240	6377	6492						
SVC	680#	681													
XFER	2619#	5366#	5579#	5804#	5952#	6137#	6204#								
SMD	1845#	1859	1895	1931	1967	2002	2037	2073	2109	2144	2179	2214	2249	2284	2320
	2357	2385	2420												

. ABS. 032002 000

CZKMAO KMC11-B STATIC PART2
CZKMA.P11 20-OCT-81 17:00

MACY11 30A(1052) 21-OCT-81 09:50 F 16 PAGE 201
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0199

ERRORS DETECTED: 0

CZKMA,CZKMA/CRF/SOL/NL:TOC=SVC34R/ML,CZKMA
RUN-TIME: 243 281 31 SECONDS
RUN-TIME RATIO: 2514/557=4.5
CORE USED: 20K (39 PAGES)