

KMC11

CPU MICRO DIAGNOSTICS
MD-11-DZKCA-A

EP-DZKCA-A-DL-A
COPYRIGHT © 1977

AUG 1977
digital
MADE IN USA

FICHE 1 OF 2

This microfiche card contains 168 frames of data, organized into 14 columns and 12 rows. Each frame displays a small, high-contrast image of a document page, which appears to be a technical manual or diagnostic guide. The text within the frames is too small to be legible, but the layout suggests a structured presentation of information, possibly including diagrams, tables, and text blocks. The card is labeled 'FICHE 1 OF 2' at the top right, indicating it is the first of two pages in a set.

KMC11

CPU MICRO DIAGNOSTICS
MD-11-DZKCA-A

EP-DZKCA-A-DL-A
COPYRIGHT © 1977
FICHE 2 OF 2

AUG 1977
digital
MADE IN USA

The microfiche card displays a grid of 144 frames of diagnostic data. Each frame contains a different set of information, likely related to CPU microdiagnostics. The data is presented in a structured, tabular format within each frame. The frames are arranged in 12 rows and 12 columns. The data in the frames appears to be organized into sections, possibly representing different diagnostic tests or components. The text is small and dense, typical of microfiche storage. The overall layout is consistent across the entire card, with each frame providing a unique view of the diagnostic data.

144 frames of diagnostic data

B01

* JAKUBBSEQ
PDP10 PAGE: 0001

00010000

770720

PDP10 411

HDR1DZKCRASEQ

00010000

770720

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZKCA-A-D
PRODUCT NAME: KMC11 CPU MICRO-DIAGNOSTICS.
DATE: MAY 1977
MAINTAINER: DIAGNOSTICS
AUTHOR: DINESH GORADIA

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Digital.

Copyright (C) 1977 by Digital Equipment Corporation

1. ABSTRACT

The function of the KMC11 diagnostics is to verify that the option operates according to specifications. The diagnostics verify that there are no malfunctions and that all operations of the KMC11 are correct in its environment.

Parameters must be set up to alert the diagnostics to the KMC11 configuration. These parameters are contained in the STATUS TABLE and are generated in two ways: 1) Manual input - the operator answers questions. 2) Autosizing - the program determines the parameters automatically.

DZKCA tests the KMC11 micro-processor (MB204). It performs write/read tests on the KMC unibus registers, checks the micro-processor operation, checks out Main Memory, scratch pad memory, the ALU functions as well as interrupts and IOP operation. DZKCA performs no tests on the line unit or any CRAM dependent tests. It will run on KMC11's containing CRAM (IOP). It does not require a line unit to run.

Currently there are four off line diagnostics that are to be run in sequence to insure that if an error should occur it will be detected at an early stage.

NOTE: Additional diagnostics may be added in the future.

The four diagnostics are:

1. DZKCC [REV] Basic W/R and Micro-processor tests
2. DZKCD [REV] Jump and memory tests (Heat test tape)
3. DZKCE [REV] DDCMP Line unit tests
4. DZKCF [REV] BITSTUFF Line unit tests
5. DZKCA [REV] KMC11 CPU MICRO-DIAGNOSTICS.

2. REQUIREMENTS

2.1 EQUIPMENT

Any PDP11 family CPU (except an LSI-11) with minimum 8k memory
ASR 33 (or equivalent)
KMC11-AN IOP (MB204)

2.2 STORAGE

Program will use all BK of memory except where ABL and BOOTSTRAP LOADER reside. Locations 2100 thru 2300; contain the "STATUS TABLE" information which is generated at start of diagnostics by manual input (questions) or automatically (auto-sizing). This area is an overlay area and should not be altered by the operator.

3. LOADING PROCEEDURE

3.1 METHOD

All programs are in absolute format and are loaded using the ABSOLUTE LOADER. NOTE: if the diagnostics are on a media such as DISK, MAGTAPE, DECTAPE, or CASSETTE; follow instructions for the monitor which has been provided on that specific media.

ABSOLUTE LOADER starting address #500

MEMORY * SIZE

4k	17
8k	37
12k	57
16k	77
20k	117
24k	137
28k	157

- 3.1.1 Place address of ABS loader into switch register.
(also place 'HALT' SW up)
- 3.1.2 Depress 'LOAD ADDRESS' key on console and release.
- 3.1.3 Depress 'START KEY' on console and release (program should now be loading into CPU)

4. STARTING PROCEEDURE

- a. Set switch register to 000200
- b. Depress 'LOAD ADDRESS' key and release
- c. Set SWR to zero for 'AUTO SIZING' or SWR bit0=1 for manual input (questions) or SWR bit7=1 to use existing parameters set up by a previous start or a previously run KMC11 diagnostic.
- d. Depress 'START KEY' and release. The program will type Maindec Name and program name (if this was the first start up of the program) and also the following:

MAP OF KMC11 STATUS

<u>PC</u>	<u>CSR</u>	<u>STAT1</u>	<u>STAT2</u>	<u>STA13</u>
002100	160010	045310	177777	000000
002110	160020	045320	177777	000000

The program will type 'R' and proceed to run the diagnostic. The above is only an example. This would indicate the status table starting at add. 2100 in the program. In this example the table contains the information and status of two KMC11'S. THE STATUS TABLE MUST BE VERIFIED BY THE USER IF AUTO SIZING IS DONE. For information of status table see section 8.4 for help.

If the diagnostic was started with SW00=1 indicating manual parameter input then the following shows an example of the questions asked and some example answers:

HOW MANY KMC11'S TO BE TESTED?1

01
 CSR ADDRESS?160010
 VECTOR ADDRESS?310
 BR PRIORITY LEVEL? (4,5,6,7)?5
 WHICH LINE UNIT? IF NONE TYPE "N", IF M8201 TYPE "1", IF M8202 TYPE "2"?1
 IS THE LOOP BACK CONNECTOR ON?Y
 SWITCH PAC#1 (DOCHP LINE#)?377
 SWITCH PAC#2 (BMB73 BOOT ADD)?377

Following the questions the status map is printed out as described above, the information in the map reflects the answers to the questions. If the diagnostic was started with SW00=0 and SW07=0 (AUTO-SIZING) then no questions are asked and only the status-map is printed out. If AUTO-SIZING is used the status information must be verified to be correct (match the hardware). if it does not match the hardware the diagnostic must be restarted with SW00=1 and the questions answered.

4.1 CONTROL SWITCH SETTINGS

- SW15 Set: Halt on error
- SW14 Set: Loop on current test
- SW13 Set: Inhibit error print out
- SW12 Set: Inhibit type out abell on error.
- SW11 Set: Inhibit iterations. (quick pass)
- SW10 Set: Escape to next test on error
- SW09 Set: Loop with current data
- SW08 Set: Catch error and loop on it
- SW07 Set: Use previous status table.
- SW06 Set: Halt in ROMCLK routine before clocking micro-processor
- SW05 Set: Reserved
- SW04 Set: Reserved
- SW03 Set: Reselect KMC11's desired active
- SW02 Set: Lock on selected test
- SW01 Set: Restart program at selected test
- SW00 Set: Build new status table from questions. (If SW07=0 and SW00=0 a new status table is built by auto-sizing)

Switch 06 and 08-15 are dynamic and can be changed as needed while the diagnostic is running. Switches 00-03 and switch 07 are static, and are used only on starting or restarting the diagnostic.

4.1.2 SWITCH REGISTER OPTIONS (at start up)

SW 01 RESTART PROGRAM AT SELECTED TEST. It is strongly suggested that at least one pass has been made before trying to select a test, the reason being is that the program has to clear areas and set up parameters. When this switch is used the diagnostic will ask TEST NO.? Answer by typing the number of the test desired and carriage return to begin execution at the selected test.

SW 02 LOCK ON SELECTED TEST. This switch when used with SW01 will cause the program to constantly loop on the selected test. Hitting any key on the console will let it advance to the next test and loop until a key is hit again. If SW02=0 when SW01 is used. The program will begin at the selected test and continue normal operations.

SW 03 RESELECT KMC11'S DESIRED ACTIVE. Please note that a message is typed out for setting the switch register equal to KMC11's active. This means if the system has four KMC11s; bits 00,01,02,03 will be set in loc 'KMACTV' from the switch register. Using this switch(SW00) alters that location; therefore if four KMC11s are in the system **###00 NOT###** set switches greater than SW 03 in the up position. This would be a fatal error. do not select more active KMC11s than there is information on in the status table.

METHOD: A: Load address 200
B: Start with SW 00=1
C: Program will type message
D: Set a switch for each KMC desired active.
EXAMPLE: If you have 4 KMC's but only want to run the first and the last set SWR bits 0 and 3 = 1. PRESS CONTINUE
E: Number (IF VALID) will be in data lights (excluding 11/05)
F: Set with any other switch settings desired. PRESS CONTINUE.

4.1.3 DYNAMIC SWITCHES

ERROR SWITCHES

- | | | |
|----|------|--|
| 1. | SW12 | Delete print out/bell on error. |
| 2. | SW13 | Delete error printout. |
| 3. | SW15 | Halt on the error. |
| 4. | SW08 | Go to beginning of the test(on error). |
| 5. | SW10 | Go to next test(on error). |

SCOPE SWITCHES

1. SW06 Halt in ROMCLK routine before clocking micro-processor instruction. This allows the operator to scope a micro-processor instruction in the static state before it is clocked. Hit continue to resume running.
2. SW09 (if enabled by 'SCOPI') on an error; If an '#' is printed in front of the test no. (ex. #TEST NO. 10) SW09 is incorporated in that test and therefore SW09 is usually the best switch for the scope loop (SW14=0, SW10=0, SW09=1, SW08=0). If SW09 is not enabled; and there is a HARD error (constant); SW08 is best. (SW14=1,0, SW10=0, SW09=0, SW08=1). for intermittent errors; SW14=1 will loop on test regardless of error or not error. (SW14=1, SW10=0, SW09=0, SW08=1,0)
3. SW11 Inhibit iterations.
4. SW14 Loop on current test.

4.2 STARTING ADDRESS

Starting address is at 000200 there are no other starting addresses for the KNC11 diagnostics. (See Section 4.0)

NOTE: If address 000042 is non-zero the program assumes it is under ACT11 or XXDP control and will act accordingly after all available KNC11's are tested the program will return to 'XXDP' or 'ACT-11'.

5. OPERATING PROCEDURE

When program is initially started messages as described in section 4.0 will be printed, and program will begin running the diagnostic

5.2 PROGRAM AND/OR OPERATOR ACTION

The typical approach should be

1. Halt on error (via SW 15=1) when ever an error occurs.
2. Clear SW 15.
3. Set SW 14: (loop on this test)
4. Set SW 13: (inhibit error print out)

The TEST NUMBER and PC will be typed out and possibly an error message (this depends on the test) to give the operator an idea as to the source of the problem. If it is necessary to know more information concerning the error report; LOOK IN THE LISTING for that TEST NUMBER which was typed out and then NOTE THE PC of the ERROR REPORT this way the EXACT FUNCTION of the test CAN BE DETERMINED.

6. ERRORS

As described previously there will always be a TEST NUMBER and PC typed out at the time of an error (providing SW 13=0 and SW 12=0). in most cases additional information will be supplied in the the error message to give the operator an indication of the error.

6.2 ERROR RECOVERY

If for some reason the KMC11 should 'HANG THE BUS' (gain control of bus so that console manual functions are inhibited) an init or power down/up is necessary for operator to regain control of cpu. If this should happen; look in location 'STSTNM' (address 1202) for the number of the test that was running at the time of the catastrophic error. In this way the operator will have an idea as to what the KMC11 was doing at the time of the error.

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

See section 4. (PLEASE)
Status table should be verified regardless of how program was started. Also it is important to use this listing along with the information printed on the TTY to completely isolate problems.

7.2 OPERATING RESTRICTIONS

The first time a KMC11 diagnostic is loaded into core and run the STATUS TABLE must be set up. This is done by manual input (SM00=1) or by autosizing (SM00=0 and SM07=0). Thereafter however the status table need not be setup by subsequent restarts or even loading the next KMC diagnostic because the STATUS TABLE is overlayed. The current parameters in the STATUS TABLE are used when SM07=1 on start up.

7.3 HARDWARE CONFIGURATION RESTRICTIONS

KMC11 IOP(MB204)- Jumper W1 must be in,

8. MISCELLANEOUS

8.1 EXECUTION TIME

All KMC11 device diagnostics will give an 'END PASS' message (providing no errors and sw12=0) within 4 mins. This is assuming SW11=1 (DELETE ITERATIONS) is set to give the fastest possible execution. The actual execution time depends greatly on the PDP11 CPU configuration and the amount of memory in the system.

8.2 PASS COMPLETE

NOTE: EVERY time the program is started; the tests will run as if SW11 (delete iterations) was up (=1). This is to 'VERIFY NO HARD ERRORS' as soon as possible. Therefore the first pass -EACH TIME PROGRAM IS STARTED- will be a 'QUICK PASS' until all KMC11's in system are tested. When the diagnostic has completed a pass the following is an example of the print out to be expected.

```
END PASS DZKCA CSR: 175000 VEC: 0300 PASSES: 000001
ERRORS: 000000
```

NOTE: The pass count and error counts are cumulative for each KMC11 that is running, and are set to zero only when the diagnostic is started. Therefore after an overnight run for example, the total passes and errors for each KMC11 since the diagnostic was started are reflected in PASSES: and ERRORS:.

8.4 KEY LOCATIONS

- SLPADR (1206) Contains the address where program will return when iteration count is reached or if loop on test is asserted.
- NEXT (1442) Contains the address of the next test to be performed.
- STSTNM (1202) Contains the number of the test now being performed.
- RUN (1500) The bit in 'RUN' always points to the KMC11 currently being tested. EXAMPLE: (RUN) 1500/00000000100000 Means that KMC11 no.06 is the KMC11 now running.

KMCROO-KMCR17
KMST00-KMST17
(2100)-(2300)

These locations contain the information needed to test up to 16 (decimal) KMC11s sequentially. they contain the CSR, VECTOR and STATUS concerning the configuration of each KMC11.

- KMACTV (1306) Each bit set in this location indicates that the associated KMC11 will be tested in turn. EXAMPLE: (KMACTV) 1470/000000000011111 means that KMC11 no. 00,01,02,03,04 will be tested. EXAMPLE: (KMACTV) 1470/000000000010001 Means that KMC11 no. 00,04 will be tested.

- KMCSR (2066) Contains the CSR of the current KMC11 under test.

8.4A 'STATUS TABLE' (2100-2300)

The table is filled by AUTO SIZING or by the manual parameter input (questions) as described previously. Also if desired by user; the locations may be altered by hand (toggled in) to suit the specific configuration.

The example status map shown below contains information for two KMC11'S. the table can contain up to 16 KMC11'S. Following the map is a description of the bits for each map entry

MAP OF KMC11 STATUS

PC	CSR	STAT1	STAT2	STAT3
002100	160010	045310	177777	000000
002110	160020	016320	000000	000000

Each map entry contains 4 words which contain the status information for 1 KMC11. The PC shows where in core memory the first of the 4 words is. In the example above the first KMC'S status is in locations, 2100, 2102, 2104, and 2106. The second KMC status is located at 2110, 2112, 2114, and 2116. The information contained in each 4 word entry is defined as follows:

CSR: Contains KMC11 CSR address

STAT1: BITS 00-08 IS KMC11 VECTOR ADDRESS
BIT14=1 TURNAROUND CONNECTOR IS ON
BIT14=0 NO TURNAROUND CONNECTOR
BIT13=0 LINE UNIT IS AN M8201
BIT13=1 LINE UNIT IS AN M8202
BIT12=1 NO LINE UNIT
BITS 09-11 IS KMC11 BR PRIORITY LEVEL

STAT2: LOW BYTE IS SWITCH PAC#1 (DDCMP LINE NUMBER)
HIGH BYTE IS SWITCH PAC#2 (BM873 BOOT ADD)

STAT3: NOT USED

8.5 METHOD OF AUTO SIZING

8.5.1 FINDING THE CONTROL STATUS REGISTER.

The auto-sizing routine finds a KMC11 as follows: It starts at address 160000 and tests all address in increments of 10 up to and including address 167760. If the address does not time out, the following is done, the first CROM address is written to a 125252 then it is read back. If it contains a -1 or 125252 a KMC11 has been found, if not, the address is updated by 10 and the search continues. A -1 indicates a KMC11 with no CROM, and a 125252 indicates a KMC11 with CROM. Further tests are performed at this point to determine which line unit, if any, is installed, if a loop-back connector is installed and various switch settings on the line unit. THIS IS WHY THE STATUS TABLE MUST BE VERIFIED BY THE USER AND IF ANY OF THE INFORMATION DOES NOT AGREE WITH THE HARDWARE THE DIAGNOSTIC MUST BE RESTARTED AND THE QUESTIONS MUST BE ANSWERED. All KMC11's in the system will be found by the auto-sizer. If it does not find a KMC11 the diagnostic must be restarted and the questions answered.

8.5.2 FINDING THE VECTOR AND BR LEVEL

The vector area (address 300-776) is filled with the instruction IOT and '+2' (next address). The processor status is started at 7 and the KMC is programmed to interrupt. The PS is lowered by 1 until the KMC interrupts, a delay is made and if no interrupt occurs at PS level 3 (because of a bad KMC11) the program assumes vector address 300 at BR level 5 and the problem should be fixed in the diagnostic. Once the problem is fixed; the program should be re-setup again to get correct vector. If an interrupt occurred; the address to which the KMC11 interrupted to is picked up and reported as the vector. NOTE: if the vector reported is not the vector set up by you; there is a problem and AUTO SIZING should not be done.

8.5 SOFTWARE SWITCH REGISTER

If the diagnostic is run on an 11/04 or other CPU without a switch register then a software switch register is used to allow user the same switch options as described previously. If the hardware switch register does not exist or if one does and it contains all ones (177777) this software switch register is used.

Control:

To obtain control at any allowable time during execution of the diagnostic the operator types a CTRL G on the console terminal keyboard. As soon as the CTRL G is recognized, by the diagnostic, the following message will be displayed:

SMR=XXXXXX NEW?

Where XXXXXX is the current contents of the software switch register in octal. The software control routine will then await operator action. At which time the operator is required to type one or more of the legal characters: 1) 0 - 7, 2) line feed(<LF>), 3) carriage return(<CR>), or 4) control-U (CTRL U). No check is made for legality. If the input character is not a <LF>, <CR>, or CTRL U it is assumed to be an octal digit.

To change the contents of the SSR the operator simply types the new desired value in octal - leading zeros need not be typed. And terminates the input string with a <CR> or <LF> depending on the program action desired as described below. The input value will be truncated to the last 6 digits typed. At least one digit must be typed on any given input string prior to the terminator before a change to the SSR will occur.

When the input string is terminated with a <CR> the diagnostic will continue execution from the point at which it was interrupted. If a <CR> is the only thing typed the program will continue without changing the SSR. The <LF> differs from the <CR> by restarting the program as if it were restarted at address 200.

If a CTRL U is typed at any point in the input string prior to the terminator the input value will be disregarded and the prompt displayed (SWR = XXXXXX NEW?).

To set the SSR for the starting switches, first load the diagnostic, then hit CTRL G, then start the diagnostic.

APT/ACT/XODP/SLIDE

THIS DIAGNOSTIC IS APT/ACT/XODP/SLIDE COMPATIBLE USER WOULD BE ABLE TO RUN IT UNDER APT/ACT/XODP ENVIRONMENT.

NOTE: FOR MANUFACTURING PURPOSE ONLY ITS DESCRIBED HOW TO RUN UNDER APT ENVIRONMENT.

ETABLE SETTING FOR APT TO RUN UNDER APT

FIRST PASS TIME:

LONGEST TEST TIME:

ADDITIONAL TEST TIME:

ALL THE ABOVE PARAMETERS ARE DEPENDENT ON PARTICULAR DIAGNOSTICS AND SHOULD BE LOADED AT THE TIME OF SETTING ETABLE. THERE IS NO DEFAULT TIME SET UP.

SOFTWARE ENVIRONMENT:001 ENVIRONMENT MODE:200

SWITCH 1:-SHOULD BE USED AS NORMAL SWITCH REGISTER.

SWITCH 2:-NOT USED.

CPU OPTIONS:-NOT USED.

MEMORY TYPE 1:-BITS<2:4>:=BITS <12:14> OF STAT1 OF DEV:0.

MAXIMUM ADDRESS:-BITS<17:19>:=BITS<12:14> OF STAT1 OF DEV:1

BITS<2:4>:=BITS <12:14> OF STAT1 OF DEV:2

BITS<10:12>:=BITS<12:14> OF STAT1 OF DEV:3

IN THE SAME MANNER

MEMORY TYPE 2 MAXIMUM ADDRESS:-GETS STAT1<12:14> OF DEVICE 4,5,6,7.

MEMORY TYPE 3 MAXIMUM ADDRESS:-GETS STAT1<12:14> OF DEVICE 8,9,10,11.

MEMORY TYPE 4 MAXIMUM ADDRESS:-GETS STAT1<12:14> OF DEVICE 12,13,14,15.

INTERRUPT VECTOR 1:FIRST DEVICE RECEIVE VECTOR.

REST OF THE DEVICE(KMC'S) VECTOR SHOULD BE SET UP SEQUENTIALLY
IN INCREMENTS OF 10.

BUS PRIORITY:KMC'S PRIORITY(SHOULD BE SAME FOR ALL KMC'S UNDER
TEST).

INTERRUPT VECTOR 2:NOT USED.

BUS PRIORITY:NOT USED.

BASE ADDRESS:FIRST DEVICE CSR ADDRESS.

REST SHOULD FOLLOW SEQUENTIALLY
IN INCREMENTS OF 10.

DEVICE MAP:AS DESCRIBED IN APT MANUAL.

CONTROLLER SPECIFIC CODE 1:-NO. OF DEVICES UNDER TEST.

CONTROLLER SPECIFIC CODE 2:-NOT USED.

DEVICE DESCRIPTOR WORD 0:STAT2 OF FIRST DEVICE.

. .

. .

TO

. .

. .

DEVICE DESCRIPTOR WORD 15:STAT2 OF 16TH DEVICE.(KMC)

FOR KMC11 CPU MICRO-DIAGNOSTICS ONLY ::

***** THIS
DIAGNOSTIC TESTS ARE STRUCTURED IN FOLLOWING MANNER

1. SAVE CURRENT TEST NO AND ADDRESS OF NEXT TEST AS USUALLY DONE.
2. LOAD THE MICRO-CODE FOR THE PARTICULAR TEST. VERIFY WHAT WAS LOADED. AND THEN WAIT FOR THE TEST TO INTERRUPT IN SPECIFIC TIME.
3. IF THE TEST DOES NOT INTERRUPT IN SPECIFIC TIME (IN WHICH IT SHOULD HAVE COMPLETED) THEN FLAG IT AS TIME OUT ERROR AND START THE NEXT TEST. (NO LOOP ON ERROR OR SCOPE ON THIS CONDITION.)
4. IF THE MICRO-PROCESSOR PC LOGIC HAS FAULT AND IT GOES OUT OF TEST BOUNDARY THEN "UPC(MICRO-PROCESSOR PC) SEQUENCE ERROR " WOULD BE REPORTED. HERE AGAIN LOOP ON ERROR IS NOT SUPPORTED HOWEVER DIAGNOSTIC SCOPE FEATURE IS SUPPORTED. WHEN DOING NEXT ITERATION IT WOULD START THIS TEST FROM BEGINING.
5. IF ERROR ON LOADING MICRO-CODE, IT WOULD REPORT ERROR AND HALT- FOR FURTHER ACTION FROM OPERATOR.

AT THIS POINT

STSTNM=WILL HAVE THE TEST NO. AT WHICH ERROR OCCURED.

RD=POINTS TO GOOD MICRO-CODE INSTRUCTION.

R1=HAS CSR ADDRESS OF KMC UNDER TEST.

R2=POINTS THE CRAM LOCATION.

6. ALL THE REGULAR DATA ERRORS AND HARDWARE ERRORS ARE REPORTED WITH

TEST NO: XXX PC: STARTING ADDRESS OF THE TEST.

" ERROR MESSAGE "

GOOD BAD UPC XXXXXX(IF REQUIRED)

WITH ABOVE INFORMATION USER CAN LOCATE ERROR OCCURENCE IN PROGRAM LISTING AT TEST NO: XXX OR PC=START ADDRESS OF THE TEST. THEN LOOKING FOR MICRO-PROCESSOR PC(MICPC)=UPC.

*

DOCUMENT

MAINDEC-11-DZKCA-A

COPYRIGHT 1977
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS. 01754

2262 ***** TEST 1 *****
 MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
 FLOAT A 1 THROUGH REGISTER OUT1 <4>
 FLOAT A 0 THROUGH REGISTER OUT1 <4>

2459 ***** TEST 2 *****
 MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
 FLOAT A 1 THROUGH REGISTER OUT1 <5>
 FLOAT A 0 THROUGH REGISTER OUT1 <5>

2656 ***** TEST 3 *****
 MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
 FLOAT A 1 THROUGH REGISTER OUT1 <6>
 FLOAT A 0 THROUGH REGISTER OUT1 <6>

2853 ***** TEST 4 *****
 MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
 FLOAT A 1 THROUGH REGISTER OUT1 <7>
 FLOAT A 0 THROUGH REGISTER OUT1 <7>

3050 ***** TEST 5 *****
 MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
 FLOAT A 1 THROUGH REGISTER OUT1 <10>
 FLOAT A 0 THROUGH REGISTER OUT1 <10>
 THE NPR RD BIT (BIT0) IS MASKED DURING THIS TEST.

3266 ***** TEST 6 *****
 MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
 FLOAT A 1 THROUGH REGISTER OUT1 <11>
 FLOAT A 0 THROUGH REGISTER OUT1 <11>
 THE BR RD BIT, PGM CLOCK BIT, FORCE POWER FAIL BIT
 (BITS 7,4,1) ARE ALL MASKED DURING THIS TEST

3501 ***** TEST 7 *****
 MICRO PROCESSOR OUT0 REGISTER WRITE/READ TEST.
 FLOAT A 1 THROUGH REGISTER OUT0 <0>
 FLOAT A 0 THROUGH REGISTER OUT0 <0>

- 3698 ***** TEST 10 *****
MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
- 3700 FLOAT A 1 THROUGH REGISTER OUTO <1>
 FLOAT A 0 THROUGH REGISTER OUTO <1>
- 3895 ***** TEST 11 *****
MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
 FLOAT A 1 THROUGH REGISTER OUTO <2>
 FLOAT A 0 THROUGH REGISTER OUTO <2>
- 4092 ***** TEST 12 *****
MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
 FLOAT A 1 THROUGH REGISTER OUTO <3>
 FLOAT A 0 THROUGH REGISTER OUTO <3>
- 4289 ***** TEST 13 *****
MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
 FLOAT A 1 THROUGH REGISTER OUTO <4>
 FLOAT A 0 THROUGH REGISTER OUTO <4>
- 4486 ***** TEST 14 *****
MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
 FLOAT A 1 THROUGH REGISTER OUTO <5>
 FLOAT A 0 THROUGH REGISTER OUTO <5>
- 4683 ***** TEST 15 *****
MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
 FLOAT A 1 THROUGH REGISTER OUTO <6>
 FLOAT A 0 THROUGH REGISTER OUTO <6>
- 4880 ***** TEST 16 *****
MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
 FLOAT A 1 THROUGH REGISTER OUTO <7>
 FLOAT A 0 THROUGH REGISTER OUTO <7>
- 5077 ***** TEST 17 *****
MICRO PROCESSOR B REGISTER TEST
 FLOAT A 1 THROUGH THE BREG.
 FLOAT A 0 THROUGH THE BREG.
- 5255 ***** TEST 20 *****
SCRATCH PAD TEST FOR SP4
 FLOAT A 1 THROUGH SCRATCH PAD 4
 FLOAT A 0 THROUGH SCRATCH PAD 4
- 5462 ***** TEST 21 *****
SCRATCH PAD TEST FOR SP5
 FLOAT A 1 THROUGH SCRATCH PAD 5
 FLOAT A 0 THROUGH SCRATCH PAD 5

- 5669 ***** TEST 22 *****
 SCRATCH PAD TEST FOR SP6
 FLOAT A 1 THROUGH SCRATCH PAD 6
 FLOAT A 0 THROUGH SCRATCH PAD 6
- 5876 ***** TEST 23 *****
 SCRATCH PAD TEST FOR SP7
 FLOAT A 1 THROUGH SCRATCH PAD 7
 FLOAT A 0 THROUGH SCRATCH PAD 7
- 6083 ***** TEST 24 *****
 SCRATCH PAD TEST FOR SP10
 FLOAT A 1 THROUGH SCRATCH PAD 10
 FLOAT A 0 THROUGH SCRATCH PAD 10
- 6290 ***** TEST 25 *****
 SCRATCH PAD TEST FOR SP11
 FLOAT A 1 THROUGH SCRATCH PAD 11
 FLOAT A 0 THROUGH SCRATCH PAD 11
- 6497 ***** TEST 26 *****
 SCRATCH PAD TEST FOR SP12
 FLOAT A 1 THROUGH SCRATCH PAD 12
- 6500 FLOAT A 0 THROUGH SCRATCH PAD 12
- 6704 ***** TEST 27 *****
 SCRATCH PAD TEST FOR SP13
 FLOAT A 1 THROUGH SCRATCH PAD 13
 FLOAT A 0 THROUGH SCRATCH PAD 13
- 6911 ***** TEST 30 *****
 SCRATCH PAD TEST FOR SP14
 FLOAT A 1 THROUGH SCRATCH PAD 14
 FLOAT A 0 THROUGH SCRATCH PAD 14
- 7118 ***** TEST 31 *****
 SCRATCH PAD TEST FOR SP15
 FLOAT A 1 THROUGH SCRATCH PAD 15
 FLOAT A 0 THROUGH SCRATCH PAD 15
- 7325 ***** TEST 32 *****
 NPR TEST
 TEST OF DATI, 1 WORD FROM 11 MEMORY TO UPROC
- 7498 ***** TEST 33 *****
 NPR TEST
 TEST OF DATO, 1 WORD FROM UPROC TO PDP11 MEMORY.
 THEN DATI OF THAT WORD AND CHECK....

7741 ***** TEST 34 *****
 NPR NON-EXISTENT MEMORY TEST
 DO A DATO TO A NON-EXISTENT ADDRESS.
 VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11

7877 ***** TEST 35 *****
 NPR TEST
 TEST OF MULTIPLE NPR'S DOING DATI.
 XFER 6 WORD'S FROM 11 MEMORY TO UPROC.

8119 ***** TEST 36 *****
 MAIN MEMORY TEST
 FLOAT A 0 THROUGH ALL MAIN MEMORY LOCATION...

8276 ***** TEST 37 *****
 MAIN MEMORY TEST
 FLOAT A 1 THROUGH ALL MAIN MEMORY LOCATIONS...

8436 ***** TEST 40 *****
 MAIN MEMORY DUAL ADDRESSING TEST
 LOAD EACH MEMORY LOCATION WITH IT'S OWN PAGE LESS ADDRESS.
 READ BACK EACH LOCATION TO VERIFY CORRECT ADDRESSING.

8649 ***** TEST 41 *****
 MAR TEST.
 PERFORM DUAL ADDRESSING TEST.
 USING MAR AUTO-INC FEATURE.

8815 ***** TEST 42 *****
 ALU C BIT TEST
 TEST THAT ADD OF 377 AND 1 WILL SET TAG CBIT.
 THEN CHECK IF C BIT CLEARS

8938 ***** TEST 43 *****
 ALU TEST
 TEST OF ALU FUNCTION SEL B & SEL A WITH C BIT CLEARED.
 TEST OF ALU FUNCTION SEL B & SEL A WITH C BIT SET.
 ALU FUNCTION (B)
 LOAD MAIN MEMORY 16 WORDS OF DATA.
 PERFORM THE FUNCTION, VERIFY THE RESULTS..

9255 ***** TEST 44 *****
 ALU TEST
 TEST OF ALU FUNCTION A OR NOTB WITH C BIT CLEARED.
 TEST OF ALU FUNCTION A OR NOTB WITH C BIT SET.
 ALU FUNCTION (A OR NOTB)
 LOAD MAIN MEMORY 16 WORDS OF DATA.
 PERFORM THE FUNCTION, VERIFY THE RESULTS..

```

9572 ***** TEST 45 *****
ALU TEST
TEST OF ALU FUNCTION A AND B WITH C BIT CLEARED.
TEST OF ALU FUNCTION A AND B WITH C BIT SET.
ALU FUNCTION (A AND B)
LOAD MAIN MEMORY 16 WORDS OF DATA.
PERFORM THE FUNCTION, VERIFY THE RESULTS..

9889 ***** TEST 46 *****
ALU TEST
TEST OF ALU FUNCTION A OR B WITH C BIT CLEARED.
TEST OF ALU FUNCTION A OR B WITH C BIT SET.
ALU FUNCTION (A OR B)
LOAD MAIN MEMORY 16 WORDS OF DATA.
PERFORM THE FUNCTION, VERIFY THE RESULTS..

10206 ***** TEST 47 *****
ALU TEST
TEST OF ALU FUNCTION A XOR B WITH C BIT CLEARED.
TEST OF ALU FUNCTION A XOR B WITH C BIT SET.
ALU FUNCTION (A XOR B)
LOAD MAIN MEMORY 16 WORDS OF DATA.
PERFORM THE FUNCTION, VERIFY THE RESULTS..

10523 ***** TEST 48 *****
ALU TEST
TEST OF ALU FUNCTION A AND B WITH C BIT CLEARED.
TEST OF ALU FUNCTION A AND B WITH C BIT SET.
ALU FUNCTION (A AND B)
LOAD MAIN MEMORY 16 WORDS OF DATA.
PERFORM THE FUNCTION, VERIFY THE RESULTS..

10840 ***** TEST 49 *****
ALU TEST
TEST OF ALU FUNCTION A OR B WITH C BIT CLEARED.
TEST OF ALU FUNCTION A OR B WITH C BIT SET.
ALU FUNCTION (A OR B)
LOAD MAIN MEMORY 16 WORDS OF DATA.
PERFORM THE FUNCTION, VERIFY THE RESULTS..

11157 ***** TEST 50 *****
ALU TEST
TEST OF ALU FUNCTION A XOR B WITH C BIT CLEARED.
TEST OF ALU FUNCTION A XOR B WITH C BIT SET.
ALU FUNCTION (A XOR B)
LOAD MAIN MEMORY 16 WORDS OF DATA.
PERFORM THE FUNCTION, VERIFY THE RESULTS..

11474 ***** TEST 53 *****
ALU TEST
TEST OF ALU FUNCTION A AND B WITH C BIT CLEARED.
TEST OF ALU FUNCTION A AND B WITH C BIT SET.
ALU FUNCTION (A AND B)
LOAD MAIN MEMORY 16 WORDS OF DATA.

```


PERFORM THE FUNCTION, VERIFY THE RESULTS..

- 11791 ***** TEST 54 *****
 ALU TEST
 TEST OF ALU FUNCTION SUB W/C WITH C BIT CLEARED.
 TEST OF ALU FUNCTION SUB W/C WITH C BIT SET.
 ALU FUNCTION (A-B-C)
 LOAD MAIN MEMORY 16 WORDS OF DATA.
 PERFORM THE FUNCTION, VERIFY THE RESULTS..

- 12108 ***** TEST 55 *****
 ALU TEST
 TEST OF INC A WITH C BIT CLEARED.
 TEST OF INC A WITH C BIT SET.
 ALU FUNCTION (A PLUS 1)
 LOAD MAIN MEMORY 16 WORDS OF DATA.
 PERFORM THE FUNCTION, VERIFY THE RESULTS..

- 12425 ***** TEST 56 *****
 ALU TEST
 TEST OF ALU FUNCTION 2A WITH C BIT CLEARED.
 TEST OF ALU FUNCTION 2A WITH C BIT SET.
 ALU FUNCTION (A PLUS A)
 LOAD MAIN MEMORY 16 WORDS OF DATA.
 PERFORM THE FUNCTION, VERIFY THE RESULTS..

- 12742 ***** TEST 57 *****
 ALU TEST
 TEST OF ALU FUNCTION A PLUS C WITH C BIT CLEARED.
 TEST OF ALU FUNCTION A PLUS C WITH C BIT SET.
 ALU FUNCTION (A PLUS C)
 LOAD MAIN MEMORY 16 WORDS OF DATA.
 PERFORM THE FUNCTION, VERIFY THE RESULTS..

- 13059 ***** TEST 60 *****
 ALU TEST
 TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT CLEARED.
 TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT SET.
 ALU FUNCTION (A-B-1)
 LOAD MAIN MEMORY 16 WORDS OF DATA.
 PERFORM THE FUNCTION, VERIFY THE RESULTS..

- 13376 ***** TEST 61 *****
 ALU TEST
 TEST OF ALU FUNCTION DEC A WITH C BIT CLEARED.
 TEST OF ALU FUNCTION DEC A WITH C BIT SET.
 ALU FUNCTION (A-1)
 LOAD MAIN MEMORY 16 WORDS OF DATA.
 PERFORM THE FUNCTION, VERIFY THE RESULTS..

13693 ***** TEST 62 *****
 TEST OF PROGRAM CLOCK BIT
 DO A MASTER CLEAR, VERIFY THAT PROGRAM CLOCK IS SET
 WRITE PROGRAM CLOCK BIT TO A ONE, VERIFY THAT IT CLEARS.
 AND THEN SETS SOME TIME LATER.

13834 ***** TEST 63 *****
 MICRO-PROCESSOR NOISE TEST.

13836 WRITE ALL ZERO'S THEN ALL ONE'S THEN A DATA PATTERN TO
 THE IBUS, TEST SP. & MAIN MEMORY
 THEN GO BACK THE DATA PATTERN
 TO VERIFY READING AND WRITING OF OTHER
 LOCATIONS. DO NOT CHANGE DATA.

14461 ***** TEST 64 *****
 HELL RAISED TEST...
 ONLY TO TEST MPRAM CONTROL LOGIC...
 NOT FOR MAINTENANCE PURPOSE...

14677 ***** TEST 65 *****
 FORCE POWER FAIL TEST.
 SET FORCE POWER FAIL VERIFY THAT PROCESSOR TRAPS TO LOC 24.
 GOING DOWN AND COMING UP. VERIFY ALSO THAT BUS INIT WAS
 BLOCKED FROM GETTING TO KMC DURING THE POWER FAIL .

14797 *****
 ERROR REPORT ROUTINE. INTERRUPTS AT LOCATION X00.
 1) CSR4:=GOOD DATA 2) CSR5:=BAD DATA
 3) CSR3:=ERROR TYPE 4) CSR6:=ERROR PC IN MICRO-CODE.
 5) CSR7:=MISCELLANEOUS INFORMATION.

30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85

000002
000002

000002

001200

000011
000012
000015
000200
177776

177774
177772
177570
177570

```

KHEMT 1?MAINDEC-11-DZKCA-A? DZKCA 1?BASIC KMC11 CONTROLLER TEST?
HEADER (MAINDEC-11-DZKCA-A),1976,(DINESH GORADIA)
TITLE MAINDEC-11-DZKCA-A
*COPYRIGHT (C) 1976
*DIGITAL EQUIPMENT CORP.
*MAYNARD, MASS. 01754
*
*PROGRAM BY DINESH GORADIA
*
*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
*PACKAGE (MAINDEC-11-DZKAC-C3), JAN 19, 1977.
*
SHEADER (1976),1?MAINDEC-11-DZKCA-A?,(DZKCA),(KMC11),1?BASIC KMC11 CONTROLLER TEST?

;-----
;MAINDEC-11-DZKCA-A BASIC KMC11 CONTROLLER TEST
;COPYRIGHT 1976, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
;-----

;STARTING PROCEDURE
;LOAD PROGRAM
;LOAD ADDRESS 000200
;SMA=0 AUTOSIZE KMC11
;SM07=1 USE CURRENT KMC11 PARAMETERS
;SM00=1 INPUT NEW KMC11 PARAMETERS
;PRESS START
;PROGRAM WILL TYPE "MAINDEC-11-DZKCA-A BASIC KMC11 CONTROLLER TEST"
;PROGRAM WILL TYPE STATUS MAP
;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
;AND THEN RESUME TESTING
;SUBSEQUENT RESTARTS WILL NOT TYPE PROGRAM TITLE

.SBTTL BASIC DEFINITIONS

;#INITIAL ADDRESS OF THE STACK POINTER *** 1200 ***
STACK= 1200
.EQUIV ENT,ERROR ;;BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE ;;BASIC DEFINITION OF SCOPE CALL

;#MISCELLANEOUS DEFINITIONS
HT= 11 ;;CODE FOR HORIZONTAL TAB
LF= 12 ;;CODE FOR LINE FEED
CR= 15 ;;CODE FOR CARRIAGE RETURN
CRLF= 200 ;;CODE FOR CARRIAGE RETURN-LINE FEED
PS= 177776 ;;PROCESSOR STATUS WORD
.EQUIV PS,PSW
STKLM= 177774 ;;STACK LIMIT REGISTER
PIRQ= 177772 ;;PROGRAM INTERRUPT REQUEST REGISTER
DSWA= 177570 ;;HARDWARE SWITCH REGISTER
DDISP= 177570 ;;HARDWARE DISPLAY REGISTER

;#GENERAL PURPOSE REGISTER DEFINITIONS

```


BASIC DEFINITIONS

142 001000
143 000400
144 000200
145 000100
146 000040
147 000020
148 000010
149 000004
150 000002
151 000001

BIT09= 1000
BIT08= 400
BIT07= 200
BIT06= 100
BIT05= 40
BIT04= 20
BIT03= 10
BIT02= 4
BIT01= 2
BIT00= 1
.EQUIV BIT09,BIT9
.EQUIV BIT08,BIT8
.EQUIV BIT07,BIT7
.EQUIV BIT06,BIT6
.EQUIV BIT05,BIT5
.EQUIV BIT04,BIT4
.EQUIV BIT03,BIT3
.EQUIV BIT02,BIT2
.EQUIV BIT01,BIT1
.EQUIV BIT00,BIT0

164 000004
165 000010
166 000014
167 000014
168 000014
169 000020
170 000024
171 000030
172 000034
173 000060
174 000064
175 000240

;;BASIC "CPU" TRAP VECTOR ADDRESSES
ERRVEC= 4 ;; TIME OUT AND OTHER ERRORS
RESVEC= 10 ;; RESERVED AND ILLEGAL INSTRUCTIONS
TBITVEC= 14 ;; T BIT
TRAVEC= 14 ;; TRACE TRAP
BPTVEC= 14 ;; BREAKPOINT TRAP (BPT)
IOTVEC= 20 ;; INPUT/OUTPUT TRAP (IOT) **SCOPE**
PWAVEC= 24 ;; POWER FAIL
EMTVEC= 30 ;; EMULATOR TRAP (EMT) **ERROR**
TRAPVEC= 34 ;; "TRAP" TRAP
TKVEC= 60 ;; TTY KEYBOARD VECTOR
TPVEC= 64 ;; TTY PRINTER VECTOR
PIRQVEC= 240 ;; PROGRAM INTERRUPT REQUEST VECTOR

;;INSTRUCTION DEFINITIONS

183 005746
184 005726
185 010046
186 012600
187 024646
188 022626
190
191
192

PUSH1SP=5746 ;; DECREMENT PROCESSOR STACK 1 WORD
POP1SP=5726 ;; INCREMENT PROCESSOR STACK 1 WORD
PUSHRD=10046 ;; SAVE RD ON STACK
POPRD=12600 ;; RESTORE RD FROM STACK
PUSH2SP=24646 ;; DECREMENT STACK TWICE
POP2SP=22626 ;; INCREMENT STACK TWICE
.EQUIV EMT,HLT ;; BASIC DEFINITION OF ERROR CALL

296
297
298
299
300 001316 000000
301 001316 000000
302 001320 000000
303 001322 000000
304 001324 000000
305 001326 000000
306 001330 000000
307 001332 000000
308 001334 000000
309 001336 002
310 001337 000
311 001340 000000
312 001342 000000
313 001344 000000
314
315
316
317
318
319
320 001346 000
321 001347 000
322
323
324
325
326 001350 000000
327
328 001352 000
329 001353 000
330 001354 000000
331 001355 000
332 001356 000
333 001357 000000
334 001358 000
335 001359 000000
336 001360 000000
337 001361 000000
338 001362 000000
339 001363 000000
340 001364 000000
341 001365 000000
342 001366 000000
343 001367 000000
344 001368 000000
345 001369 000000
346 001370 000000
347 001371 000000
348 001372 000000
349 001373 000000
350 001374 000000
351 001375 000000

```

*****
.EVEN
SMAIL:
MSGTY: .WORD  MSGTY
SFATAL: .WORD  AFATAL
STESTN: .WORD  ATESTN
SPASS: .WORD  APAS
SDEVCT: .WORD  ADEV
SUNIT: .WORD  AUNIT
SMSCAD:
MSGLEN:
SETABLE
SENV: .BYTE  AENV
SENVH: .BYTE  AENVH
SSWREG: .WORD  ASWREG
SUSWR: .WORD  AUSWR
SCPUOP: .WORD  ACPUOP
*****
: APT MAILBOX
: MESSAGE TYPE CODE
: FATAL ERROR NUMBER
: TEST NUMBER
: ASS COUNT
: EVICE COUNT
: 70 BIT NUMBER
: ADDRESS
: LENGTH
: IRONMENT TABLE
: ENVIRONMENT BYTE
: ENVIRONMENT MODE BITS
: APT SWITCH REGISTER
: USER SWITCHES
: CPU TYPE, OPTIONS
: BITS 15-11=CPU TYPE
: 11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
: 11/70=06, P00=07, 0=10
: BIT 10=REAL TIME CLOCK
: BIT 9=FLOATING POINT PROCESSOR
: BIT 8=MEMORY MANAGEMENT
: HIGH ADDRESS M.S. BYTE
: MEM. TYPE, BLK#1
: MEM. TYPE BYTE -- (HIGH BYTE)
: 500 MSEC CORE=001
: 500 MSEC BIPOLAR=002
: 500 MSEC NOS=003
: HIGH ADDRESS BLK#1
: MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
: HIGH ADDRESS M.S. BYTE
: MEM. TYPE, BLK#2
: MEM. LAST ADDRESS, BLK#2
: HIGH ADDRESS M.S. BYTE
: MEM. TYPE, BLK#3
: MEM. LAST ADDRESS, BLK#3
: HIGH ADDRESS M.S. BYTE
: MEM. TYPE, BLK#4
: MEM. LAST ADDRESS, BLK#4
: INTERRUPT VECTOR#1, BUS PRIORITY#1
: INTERRUPT VECTOR#2, BUS PRIORITY#2
: BASE ADDRESS OF EQUIPMENT UNDER TEST
: DEVICE NP
: CONTROLLER DESCRIPTION WORD#1
: CONTROLLER DESCRIPTION WORD#2
: DEVICE DESCRIPTOR WORD#0
: DEVICE DESCRIPTOR WORD#1
: DEVICE DESCRIPTOR WORD#2
: DEVICE DESCRIPTOR WORD#3
: DEVICE DESCRIPTOR WORD#4
: DEVICE DESCRIPTOR WORD#5
: DEVICE DESCRIPTOR WORD#6
: DEVICE DESCRIPTOR WORD#7
: DEVICE DESCRIPTOR WORD#8

```

352 001424 000000
 353 001426 000000
 354 001430 000000
 355 001432 000000
 356 001434 000000
 357 001436 000000
 358 001440 000000
 359
 360
 361 001442
 362
 363
 364
 365
 366 001442 000000
 367 001444 000000
 368
 369
 370
 371 001446 000000
 372 001450 000000
 373 001452 000000
 374 001454 000000
 375 001456 000000
 376 001460 000000
 377 001462 000000
 378 001464 000001
 379 001466 000000
 380 001470 000001
 381 001472 000001
 382 001474 000001
 383 001476 000001
 384 001500 000000
 385
 386 001502 002072
 387 001504 002276
 388
 389
 390
 391 001506 000
 392 001510 001510
 393 001510 000
 394 001511 000
 395
 396

SDOW9: .WORD ADDR9 ;: DEVICE DESCRIPTOR WORD#9
 SDOW10: .WORD ADDR10 ;: DEVICE DESCRIPTOR WORD#10
 SDOW11: .WORD ADDR11 ;: DEVICE DESCRIPTOR WORD#11
 SDOW12: .WORD ADDR12 ;: DEVICE DESCRIPTOR WORD#12
 SDOW13: .WORD ADDR13 ;: DEVICE DESCRIPTOR WORD#13
 SDOW14: .WORD ADDR14 ;: DEVICE DESCRIPTOR WORD#14
 SDOW15: .WORD ADDR15 ;: DEVICE DESCRIPTOR WORD#15

SETEND:

PROGRAM CONTROL PARAMETERS

 NEXT: .WORD 0 ; ADDRESS OF NEXT TEST TO BE EXECUTED
 LOCK: .WORD 0 ; ADDRESS FOR LOCK CURRENT DATA

PROGRAM VARIABLES

 STRTSM: .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
 MEHLIM: .WORD 0 ; HIGHEST LOCATION FOR NPR'S
 KMACTV: .BLKH 1 ; KMCI1 SELECTED ACTIVE
 KNUM: .BLKH 1 ; OCTAL NUMBER OF KMCI1'S
 SAVACT: .BLKH 1 ; ORIGINAL ACTIVE DEVICES.
 SAVNUM: .BLKH 1 ; WORKABLE NUMBER.
 RUN: .WORD 0 ; POINTER TO RUNNING DEVICES
 .EVEN
 CREAM: .WORD KM.MAP-6 ; TABLE POINTER
 MILK: .WORD CNT.MAP-4 ; TABLE POINTER

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 ; ON FIRST PASS OF EACH KMCI1 ITERATIONS WILL BE SUPPRES

ERROR POINTER TABLE

.SBTTL ERROR POINTER TABLE

;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
;*LOCATION SITEMB, THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
;*NOTE1: IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (SERRTPC).
;*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

;* EM ;: POINTS TO THE ERROR MESSAGE
;* DH ;: POINTS TO THE DATA HEADER
;* DT ;: POINTS TO THE DATA
;* DF ;: POINTS TO THE DATA FORMAT

SERRTB:

.EVEN

;* DF ;: DOES NOT APPLY IN THIS DIAGNOSTIC.

397
398
399
400
401
402
403
404
405
406
407
408
409
410
411 001512
412
413
414 001512 000000
415 001514 000000
416 001516 000000
417 001520 036176
418 001522 037256
419 001524 037434
420 001526 036235
421 001530 037256
422 001532 037434
423 001534 036307
424 001536 037317
425 001540 037456
426 001542 036336
427 001544 037256
428 001546 037434
429 001550 036365
430 001552 037256
431 001554 037434
432 001556 036427
433 001560 037334
434 001562 037434
435 001564 036456
436 001566 037334
437 001570 037434
438 001572 036520
439 001574 037334
440 001576 037434
441 001600 036553
442 001602 037317
443 001604 037456
444 001606 036565
445 001610 037317
446 001612 037456
447 001614 036610
448 001616 000000
449 001620 000000
450 001622 036642
451 001624 000000
452 001626 000000

DF ;: DOES NOT APPLY IN THIS DIAGNOSTIC.
0
0
EM1 ;: ERROR 1
DH1 ;: ERROR 2
DT1 ;: ERROR 3
EM2 ;: ERROR 4
DH2 ;: ERROR 5
DT2 ;: ERROR 6
EM3 ;: ERROR 7
DH3 ;: ERROR 10
DT1 ;: ERROR 11
EM11 ;: ERROR 12
DH2 ;: ERROR 13
DT2 ;: ERROR 14
EM13
0
0
EM14
0
0

453	001630	036666	EM15	
454	001632	037373	DM4	; ERROR 15
455	001634	037434	DT1	
456	001636	036732	EM16	
457	001640	000000	0	; ERROR 16
458	001642	000000	0	
459	001644	037000	EM17	
460	001646	000000	0	; ERROR 17
461	001650	000000	0	
462	001652	037030	EM20	
463	001654	037317	DM2	; ERROR 20
464	001656	037456	DT2	
465	001660	037076	EM21	
466	001662	000000	0	; ERROR 21
467	001664	000000	0	
468	001666	037126	EM22	
469	001670	000000	0	; ERROR 22
470	001672	000000	0	
471	001674	037146	EM23	
472	001676	037373	DM4	; ERROR 23
473	001700	037434	DT1	
474	001702	037210	EM24	
475	001704	000000	0	; ERROR 24
476	001706	000000	0	
477	001710	037234	EM25	
478	001712	037317	DM2	; ERROR 25
479	001714	037456	DT2	

. =2034
.SBTTL APT PARAMETER BLOCK

```

*****
;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
*****
.SX=      ;SAVE CURRENT LOCATION
.=24     ;SET POWER FAIL TO POINT TO START OF PROGRAM
200      ;FOR APT START UP
.=44     ;POINT TO APT INDIRECT ADDRESS PNTR.
SAPTHDR  ;POINT TO APT HEADER BLOCK
.=.SX    ;RESET LOCATION COUNTER
*****
;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-POPI1 DIAGNOSTIC
;INTERFACE SPEC.

```

480		002034		
481		000024		
482		000044		
483		002034		
484		002034		
485		000024		
486		000044		
487		002034		
488	000024	000024		
489		000044		
490	000044	002034		
491		002034		
492				
493				
494				
495				
496	002034			
497	002034	000000	SAPTHD:	
498	002036	001316	SHIBTS: .WORD	0 ; TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
499	002040	000132	SHEADR: .WORD	SMAIL ; ADDRESS OF APT MAILBOX (BITS 0-15)
500	002042	000137	STSTM: .WORD	90. ; RUN TIM OF LONGEST TEST
501	002044	000137	SPASTM: .WORD	95. ; RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
502	002046	000052	SUNITH: .WORD	95. ; ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
503			.WORD	SETEND-SMAIL/2 ; LENGTH MAILBOX-ETABLE(WORDS)

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

002050 000000
002052 000000
002054 000000

002056 000000
002060 000000
002062 000000
002064 000000
002066 000000
002070 000000
002072 000000
002074 000000
002076 000000

002100 000001
002102 000001
002104 000001
002106 000001

002110 000001
002112 000001
002114 000001
002116 000001

002120 000001
002122 000001
002124 000001
002126 000001

002130 000001
002132 000001
002134 000001
002136 000001

002140 000001
002142 000001
002144 000001
002146 000001

;KMC11 CONTROL INDICATORS FOR CURRENT KMC11 UNDER TEST

STAT1: 0
STAT2: 0
STAT3: 0

;KMC11 VECTOR AND REGISTER INDIRECT POINTERS

KMRVEC: 0 ; POINTER TO KMC11 RECEIVER INTERRUPT VECTOR
KMRVLV: 0 ; POINTER TO KMC11 RECEIVER INTERRUPT SERVICE PS
KMTVEC: 0 ; POINTER TO KMC11 TRANSMITTER INTERRUPT VECTOR
KMTVLV: 0 ; POINTER TO KMC11 TRANSMITTER INTERRUPT SERVICE PS
KMCSR: 0 ; POINTER TO KMC11 CONTROL STATUS REGISTER
KMCSRH: 0 ; POINTER TO KMC11 CONTROL STATUS REGISTER HIGH BYTE.
KMCTL: 0 ; POINTER TO KMC11 CONTROL OUT REGISTER
KMP04: 0 ; POINTER TO KMC11 PORT REGISTER(SEL 4)
KMP06: 0 ; POINTER TO KMC11 PORT REGISTER(SEL 6)

;TEMP STORAGE

TEMP: 0
..=..+40

;KMC11 STATUS TABLE AND ADDRESS ASSIGNMENTS

..=2100
KMRMAP: .BLKW 1
KMC00: .BLKW 1 ; CONTROL STATUS REGISTER FOR KMC11 NUMBER 00
KMS100: .BLKW 1 ; VECTOR FOR KMC11 NUMBER 00
KMS200: .BLKW 1 ; DDCMP LINE# FOR KMC11 NUMBER 00
KMS300: .BLKW 1 ; 3RD STATUS WORD

KMC01: .BLKW 1 ; CONTROL STATUS REGISTER FOR KMC11 NUMBER 01
KMS101: .BLKW 1 ; VECTOR FOR KMC11 NUMBER 01
KMS201: .BLKW 1 ; DDCMP LINE# FOR KMC11 NUMBER 01
KMS301: .BLKW 1 ; 3RD STATUS WORD

KMC02: .BLKW 1 ; CONTROL STATUS REGISTER FOR KMC11 NUMBER 02
KMS102: .BLKW 1 ; VECTOR FOR KMC11 NUMBER 02
KMS202: .BLKW 1 ; DDCMP LINE# FOR KMC11 NUMBER 02
KMS302: .BLKW 1 ; 3RD STATUS WORD

KMC03: .BLKW 1 ; CONTROL STATUS REGISTER FOR KMC11 NUMBER 03
KMS103: .BLKW 1 ; VECTOR FOR KMC11 NUMBER 03
KMS203: .BLKW 1 ; DDCMP LINE# FOR KMC11 NUMBER 03
KMS303: .BLKW 1 ; 3RD STATUS WORD

KMC04: .BLKW 1 ; CONTROL STATUS REGISTER FOR KMC11 NUMBER 04
KMS104: .BLKW 1 ; VECTOR FOR KMC11 NUMBER 04
KMS204: .BLKW 1 ; DDCMP LINE# FOR KMC11 NUMBER 04
KMS304: .BLKW 1 ; 3RD STATUS WORD

560	002150	000001	KMCR05: .BLKW	1	: CONTROL STATUS REGISTER FOR KMC11 NUMBER 05
561	002152	000001	KMS105: .BLKW	1	: VECTOR FOR KMC11 NUMBER 05
562	002154	000001	KMS205: .BLKW	1	: DDCMP LINE# FOR KMC11 NUMBER 05
563	002156	000001	KMS305: .BLKW	1	: 3RD STATUS WORD
564	002160	000001	KMCR06: .BLKW	1	: CONTROL STATUS REGISTER FOR KMC11 NUMBER 06
565	002162	000001	KMS106: .BLKW	1	: VECTOR FOR KMC11 NUMBER 06
566	002164	000001	KMS206: .BLKW	1	: DDCMP LINE# FOR KMC11 NUMBER 06
567	002166	000001	KMS306: .BLKW	1	: 3RD STATUS WORD
568	002170	000001	KMCR07: .BLKW	1	: CONTROL STATUS REGISTER FOR KMC11 NUMBER 07
569	002172	000001	KMS107: .BLKW	1	: VECTOR FOR KMC11 NUMBER 07
570	002174	000001	KMS207: .BLKW	1	: DDCMP LINE# FOR KMC11 NUMBER 07
571	002176	000001	KMS307: .BLKW	1	: 3RD STATUS WORD
572	002200	000001	KMCR10: .BLKW	1	: CONTROL STATUS REGISTER FOR KMC11 NUMBER 10
573	002202	000001	KMS110: .BLKW	1	: VECTOR FOR KMC11 NUMBER 10
574	002204	000001	KMS210: .BLKW	1	: DDCMP LINE# FOR KMC11 NUMBER 10
575	002206	000001	KMS310: .BLKW	1	: 3RD STATUS WORD
576	002210	000001	KMCR11: .BLKW	1	: CONTROL STATUS REGISTER FOR KMC11 NUMBER 11
577	002212	000001	KMS111: .BLKW	1	: VECTOR FOR KMC11 NUMBER 11
578	002214	000001	KMS211: .BLKW	1	: DDCMP LINE# FOR KMC11 NUMBER 11
579	002216	000001	KMS311: .BLKW	1	: 3RD STATUS WORD
580	002220	000001	KMCR12: .BLKW	1	: CONTROL STATUS REGISTER FOR KMC11 NUMBER 12
581	002222	000001	KMS112: .BLKW	1	: VECTOR FOR KMC11 NUMBER 12
582	002224	000001	KMS212: .BLKW	1	: DDCMP LINE# FOR KMC11 NUMBER 12
583	002226	000001	KMS312: .BLKW	1	: 3RD STATUS WORD
584	002230	000001	KMCR13: .BLKW	1	: CONTROL STATUS REGISTER FOR KMC11 NUMBER 13
585	002232	000001	KMS113: .BLKW	1	: VECTOR FOR KMC11 NUMBER 13
586	002234	000001	KMS213: .BLKW	1	: DDCMP LINE# FOR KMC11 NUMBER 13
587	002236	000001	KMS313: .BLKW	1	: 3RD STATUS WORD
588	002240	000001	KMCR14: .BLKW	1	: CONTROL STATUS REGISTER FOR KMC11 NUMBER 14
589	002242	000001	KMS114: .BLKW	1	: VECTOR FOR KMC11 NUMBER 14
590	002244	000001	KMS214: .BLKW	1	: DDCMP LINE# FOR KMC11 NUMBER 14
591	002246	000001	KMS314: .BLKW	1	: 3RD STATUS WORD
592	002250	000001	KMCR15: .BLKW	1	: CONTROL STATUS REGISTER FOR KMC11 NUMBER 15
593	002252	000001	KMS115: .BLKW	1	: VECTOR FOR KMC11 NUMBER 15
594	002254	000001	KMS215: .BLKW	1	: DDCMP LINE# FOR KMC11 NUMBER 15
595	002256	000001	KMS315: .BLKW	1	: 3RD STATUS WORD
596	002260	000001	KMCR16: .BLKW	1	: CONTROL STATUS REGISTER FOR KMC11 NUMBER 16
597	002262	000001	KMS116: .BLKW	1	: VECTOR FOR KMC11 NUMBER 16
598	002264	000001	KMS216: .BLKW	1	: DDCMP LINE# FOR KMC11 NUMBER 16
599	002266	000001	KMS316: .BLKW	1	: 3RD STATUS WORD
600	002270	000001	KMCR17: .BLKW	1	: CONTROL STATUS REGISTER FOR KMC11 NUMBER 17
601	002272	000001	KMS117: .BLKW	1	: VECTOR FOR KMC11 NUMBER 17
602	002274	000001	KMS217: .BLKW	1	: DDCMP LINE# FOR KMC11 NUMBER 17
603	002276	000001	KMS317: .BLKW	1	: 3RD STATUS WORD

L03

DZKCA MACY11 27(1006) 13-MAY-77 14:07 PAGE 15
DZKCA.P11 13-MAY-77 13:58 APT PARAMETER BLOCK
616 002300 000000 KM.END: 000000

PAGE: 0037

617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659

;KMC11 PASS COUNT AND ERROR COUNT TABLE

002302	000000	CNT.MAP:	
002302	000000	PACT00: 0	;PASS COUNT FOR KMC11 NUMBER 00
002304	000000	ERCT00: 0	;ERROR COUNT FOR KMC11 NUMBER 00
002306	000000	PACT01: 0	;PASS COUNT FOR KMC11 NUMBER 01
002310	000000	ERCT01: 0	;ERROR COUNT FOR KMC11 NUMBER 01
002312	000000	PACT02: 0	;PASS COUNT FOR KMC11 NUMBER 02
002314	000000	ERCT02: 0	;ERROR COUNT FOR KMC11 NUMBER 02
002316	000000	PACT03: 0	;PASS COUNT FOR KMC11 NUMBER 03
002320	000000	ERCT03: 0	;ERROR COUNT FOR KMC11 NUMBER 03
002322	000000	PACT04: 0	;PASS COUNT FOR KMC11 NUMBER 04
002324	000000	ERCT04: 0	;ERROR COUNT FOR KMC11 NUMBER 04
002326	000000	PACT05: 0	;PASS COUNT FOR KMC11 NUMBER 05
002330	000000	ERCT05: 0	;ERROR COUNT FOR KMC11 NUMBER 05
002332	000000	PACT06: 0	;PASS COUNT FOR KMC11 NUMBER 06
002334	000000	ERCT06: 0	;ERROR COUNT FOR KMC11 NUMBER 06
002336	000000	PACT07: 0	;PASS COUNT FOR KMC11 NUMBER 07
002340	000000	ERCT07: 0	;ERROR COUNT FOR KMC11 NUMBER 07
002342	000000	PACT10: 0	;PASS COUNT FOR KMC11 NUMBER 10
002344	000000	ERCT10: 0	;ERROR COUNT FOR KMC11 NUMBER 10
002346	000000	PACT11: 0	;PASS COUNT FOR KMC11 NUMBER 11
002350	000000	ERCT11: 0	;ERROR COUNT FOR KMC11 NUMBER 11
002352	000000	PACT12: 0	;PASS COUNT FOR KMC11 NUMBER 12
002354	000000	ERCT12: 0	;ERROR COUNT FOR KMC11 NUMBER 12
002356	000000	PACT13: 0	;PASS COUNT FOR KMC11 NUMBER 13
002360	000000	ERCT13: 0	;ERROR COUNT FOR KMC11 NUMBER 13
002362	000000	PACT14: 0	;PASS COUNT FOR KMC11 NUMBER 14
002364	000000	ERCT14: 0	;ERROR COUNT FOR KMC11 NUMBER 14
002366	000000	PACT15: 0	;PASS COUNT FOR KMC11 NUMBER 15
002370	000000	ERCT15: 0	;ERROR COUNT FOR KMC11 NUMBER 15
002372	000000	PACT16: 0	;PASS COUNT FOR KMC11 NUMBER 16
002374	000000	ERCT16: 0	;ERROR COUNT FOR KMC11 NUMBER 16
002376	000000	PACT17: 0	;PASS COUNT FOR KMC11 NUMBER 17
002400	000000	ERCT17: 0	;ERROR COUNT FOR KMC11 NUMBER 17

670
 671
 672
 673
 674
 675

FORMAT OF STATUS TABLE

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	CSR
I	C	O	N	T	R	O	L	R	E	G	I	S	T	E	R	I
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
I	*	I	*	I	*	I	*	I	*	I	*	I	*	I	*	STAT1
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
I	*	I	B	I	M	I	I	A	D	D	*	I	*	I	*	STAT2
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	*	STAT3
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	

DEFINITION OF FORMAT

- CSR: CONTAINS KMC11 CSR ADDRESS
- STAT1: BITS 00-08 IS KMC11 VECTOR ADDRESS
 BIT14=1 ??? TURNAROUND CONNECTOR IS ON
 BIT14=0 NO TURNAROUND CONNECTOR
 BIT13=0 LINE UNIT IS AN M8201
 BIT13=1 LINE UNIT IS AN M8202
 BIT12=1 NO LINE UNIT
 BITS 09-11 IS KMC11 BR PRIORITY LEVEL
- STAT2: LOW BYTE IS SWITCH PAC#1 (DCMP LINE NUMBER)
 HIGH BYTE IS SWITCH PAC#2 (BM873 BOOT ADD)
- STAT3: BIT0=1 DO FREE RUNNING TESTS ON KMC
 (MUST BE SET TO A ONE MANUALLY [PROGRAMS G AND H ONLY])

```

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

```

002402	012737	000340	177776	.START:	MOV	#340,PS	:LOCK OUT INTERRUPTS
002410	012706	001200			MOV	#STACK,SP	:SET UP STACK
002414	012737	007100	000024		MOV	#SPADR,2824	:SET UP POWER FAIL VECTOR
002418	013737	001472	001476		MOV	KONUM,SAYNUM	:SAVE NUMBER OF DEVICES IN SYSTEM.
002422	005037	011422			CLR	SWFLG	:CLEAR SOFT TYPEOUT FLAG
002426	105037	001203			CLRB	SEFLG	:CLEAR ERROR FLAG
002430	105037	001511			CLRB	OV.FLG	:ZERO QUICK VERIFY FLAG
002434	012737	002070	001502		MOV	#K1.MAP-10,CREAM	:GET MAP POINTER.
002438	012737	002276	001504		MOV	#CNT.MAP-4,MILK	:GET PASS COUNT MAP POINTER
002442	012737	100000	001500		MOV	#BIT15,RUN	:POINT POINTER TO FIRST DEVICE.
002446	012700	002302			MOV	#CNT.MAP,RO	:PASS COUNT POINTER TO RO
002450	005020			23S:	CLR	(RO)+	:CLEAR TABLE
002454	002700	002402			CHP	#CNT.MAP+100,RO	:DONE YET?
002458	005037	001216			BNE	23S	:KEEP GOING
002462	012737	000001			CLR	SEAPC	:CLEAR LAST ERROR POINTER
002466	012737	002402			MOV	#1,STSTM	:SET UP FOR TEST 1
002470	012737	002402			MOV	#.START,SLPADR	:SET UP FOR POWER FAIL BEFORE TESTING STARTS
002474							:IS IT RUNNING UNDER APT?
002478	132737	000001			BITF	#1,SENV	:IF NOT CHECK FOR TYPE OF SWITCH REGISTER.
002482	001404				BZF	3S	:LOAD SOFTWARE SWITCH REG.
002486	013737	001340			MOV	#SWREG,SWREG	:GO SET UP SOFTWARE SWITCH REG.
002490	000423				BR	6S+2	:SAVE CURRENT VECTORS
002494	013746	000006		3S:	MOV	286,-(SP)	
002498	013746	000004			MOV	284,-(SP)	
002502	012737	002606	000007		MOV	268,284	:SET UP FOR TIMEOUT
002506	012737	177570	001240		MOV	#177570,SWR	:SET SWR TO HARD SWR ADDRESS
002510	012737	177570	001242		MOV	#177570,DISPLAY	:SET DISPLAY TO HARD SWR ADDRESS
002514	022777	177777	176436		CHP	#-1,2SWR	:REFERENCE HARDWARE SWITCH REGISTER
002518	001402				BEG	6S+2	:IF = -1 USE SOFT SWR ANYWAY
002522	000407				BR	7S	:IF IT EXISTS AND NOT = -1 USE HARD SWR
002526	022626			6S:	CHP	(SP)+(SP)+	:ADJUST STACK
002530	012737	000176	001240		MOV	#SWREG,SWR	:POINTER TO SOFT SWR
002534	012737	000174	001242		MOV	#DISPREG,DISPLAY	:POINTER TO SOFT DISPLAY REG
002538	012737	000004		7S:	MOV	(SP)+,284	:RESTORE VECTORS
002542	012737	000006			MOV	(SP)+,286	
002546	105737	001506			TSTB	INIFLG	:HAS INITIALIZATION BEEN PERFORMED
002550	001006				BNE	20S	:BR IF YES
002554	022737	004070	000042		CHP	#SENDAD,2842	:IF ACT-II AUTOMATIC MODE, DON'T TYPE ID
002558	001402				BEG	20S	
002562	104401	001000			TYPE	MTITLE	:TYPE TITLE MESSAGE
002566	004737	011216		20S:	JSR	PC,CKSWR	:CHECK FOR SOFT SWR
002570	017737	176352	001446		MOV	2SWR,STRTSW	:STORE STARTING SWITCHES
002574	005737	000042			TST	2842	:IS IT RUNNING IN AUTO MODE?
002578	001402				BEG	+6	:BR IF NO
002582	005037	001446			CLR	STRTSW	:IF YES, CLEAR SWITCHES
002586	032737	000001	001446		BIT	#SM00,STRTSW	:IF SM00=1, QUESTIONS ARE ASKED.
002590	001012				BNE	17S	:BR IF SM00=1
002594	105737	001446			TSTB	STRTSW	:BIT7=1??

PROGRAM INITIALIZATION AND START UP.

```

780 002716 100007          BP      17S          ;BR IF SMO7=0
781 002720 005737 001470    TST      KNACTV      ;ARE ANY DEVICES SELECTED?
782 002724 001027          BNE     16S          ;BR IF YES
783 002726 104401 010703    TYPE,   NOACT       ;NO DEVICES SELECTED.
784 002728 000000          HALT                    ;STOP THE SHOW
785 002731 000776          BR      -2          ;DISQUALIFY CONTINUE SWITCH
786 002733 105737 001336    17S:   TSTB     SENV          ;IS IT UNDER APT DUMP MODE?
787 002735 001405          BEQ     27S          ;YES, CHECK IF APT SIZED IT?
788 002737 12737 000001 001336  BITB     B1,SENV       ;IS IT UNDER O,V OR RUN MODE?
789 002739 001012          BNE     30S          ;YES, NEEDS ONLY APT SIZING.
790 002741 000406          BR      33S          ;NO, NEEDS REGULAR AUTO.SIZE.
791 002743 105737 001337    27S:   TSTB     SENVM        ;IS IT SIZED BY APT?
792 002745 104406          BMI     30S          ;YES, NEEDS ONLY APT SIZING.
793 002747 042737 000001 001446  BIC     BSMO,STRSM    ;SIZE ONLY IN AUTO MODE.
794 002749 004737 012152    33S:   JSR      PC,AUTO.SIZE ;GO DO THE AUTO.SIZE.
795 002751 000412          BR      16S          ;GO PRINT THE MAP.
796 002753 000776          JSR      PC,APT.SIZE  ;GO DO THE APT SIZING.
797 002755 001300    16S:   TSTB     INIFLG        ;FIRST TIME?
798 002757 105737 001446    21S:   BEQ     21S          ;BR IF YES
799 002759 001301    16S:   TSTB     STRSM        ;IF USING SAME PARAMETERS DONT TYPE MAP
800 002761 100431          BMI     1S          ;
801 002763 032737 000006 001446  BIT     @BIT1:BIT2,STRSM ;IS TEST NO. OR LOCK SELECTED
802 002765 001403          BEQ     24S          ;IF NO THEN TYPE STATUS
803 002767 000424          BR      1S          ;IF YES DO NOT TYPE STATUS
804 002769 105137 001506    21S:   COMB     INIFLG        ;SET FLAG
805 002771 104401 010051    24S:   TYPE     XHEAD        ;TYPE HEADER
806 002773 012704 002100          MOV     @KCM,MAP,R4  ;SET POINTER
807 002775 010437 001276    5S:   MOV     R4,STMP0      ;SET ADDRESS
808 002777 012437 001300          MOV     (R4)+,STMP1   ;SET CSR
809 002779 001411          BEQ     1S          ;ALL DONE IF ZERO
810 002781 012437 001302          MOV     (R4)+,STMP2   ;SET STAT1
811 002783 012437 001304          MOV     (R4)+,STMP3   ;SET STAT2
812 002785 012437 001306          MOV     (R4)+,STMP4   ;SET STAT3
813 002787 104416          CONVRT                    ;TYPE OUT STATUS MAP
814 002789 011064          XSTATQ                    ;
815 002791 000762          BR      5S          ;
816 002793 003102 012700 002100    1S:   MOV     @KCM,MAP,R0  ;R0 POINTS TO STATUS TABLE
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
003106 013746 000004          MOV     @4,-(SP)      ;SAVE LOC 4
003112 013746 000006          MOV     @6,-(SP)      ;SAVE LOC 6
003116 005037 000006          CLR     @6            ;CLEAR VEC+2
003122 005037 001302          CLR     STMP2         ;CLEAR FLAG
003126 011037 002066    AUSTRT: MOV     (R0),KMC SR ;GET NEXT KMC CSR
003132 001510          BEQ     AUDONE        ;BR IF DONE
003134 012737 003240 000004  2S:   MOV     @NODEV,@4    ;SET UP FOR TIMEOUT
003142 012703 000010    3S:   MOV     @10,R3       ;R3 IS COUNT OF DEVICES BEFORE KMC

```

```

*****
*AUTO SIZE TEST
*THIS TEST VERIFYS THAT THE KMC11S AND/OR KMC11S ARE AT THE CORRECT FLOATING
*ADDRESSES FOR YOUR SYSTEM. IF THIS TEST FAILS, IT IS NOT A HARDWARE ERROR.
*CHECK THE ADDRESSES OF ALL FLOATING DEVICES (DJ,DH,DO,DU,DUP,LK,DAC,DZ,KMC).
*IF THERE ARE NO OTHER FLOATING DEVICES BEFORE THE KMC11, THE FIRST
*KMC11 IS 760110. NO DEVICE SHOULD EVER BE AT
*ADDRESS 760000.
*****

```

PROGRAM INITIALIZATION AND START UP.

836	003146	012702	003342	4S:	MOV	#DEV TAB,R2	:R2 IS DEVICE TABLE POINTER
837	003152	012701	160010		MOV	#160010,R1	:START WITH ADDRESS 160010
838	003158	005711		FLOAT:	TST	(R1)	:CHECK ADDRESS IN R1
839	003160	111204			MOV	(R2),R4	:IF NO TIMEOUT, GET NEXT ADDRESS
840	003162	060401			ADD	R4,R1	:IN R1
841	003164	005201			INC	R1	
842	003166	040401			BIC	R4,R1	
843	003170	005703			TST	R3	:ANY MORE DEVICES TO CHECK FOR?
844	003172	001371			BNE	FLOAT	:BR IF YES
845	003174	012737	003244 000004		MOV	#ERR,2#4	:OK ONLY KMC'S ARE LEFT, SET UP FOR TIMEOUT
846	003202	005711		FY:	TST	(R1)	:CHECK KMC ADDRESS
847	003204	020137	002066		CMP	R1,KMCSR	:DOES IT MATCH
848	003210	001403			BEQ	OK	:BR IF YES
849	003212	062701	000010		ADD	#10,R1	:GET NEXT KMC ADDRESS
850	003216	000771			BR	FY	:DO IT AGAIN
851	003220	062700	000010	OK:	ADD	#10,R0	:SKIP TO NEXT KMC CSR
852	003224	062701	000010		ADD	#10,R1	:GET NEXT KMC ADDRESS
853	003230	011037	002066		MOV	(R0),KMCSR	:GET NEXT KMC CSR
854	003234	001447			BEQ	AUDONE	:BRANCH IF ALL DONE.
855	003236	000761			BR	FY	:DO IT AGAIN.
856	003240	122243		NODEV:	CMPB	(R2)+,-(R3)	:ON TIMEOUT, INC R2, DEC R3
857	003242	000002			RTI		:SLPROR
858	003244	005737	001302	ERR:	TST	\$TMP2	:CHECK FLAG IF = 0 TYPE HEADER
859	003250	001014			BNE	IS	:SKIP HEADER
860	003252	104401			TYPE		:TYPEOUT HEADER MESSAGE
861	003254	010734			CONERR		:CONFIGURATION ERROR!!!!
862	003256	012737	003244 001460		MOV	#ERR,SAVPC	:SAVE PC FOR TYPEOUT
863	003264	104417			CONVRT		:TYPE OUT ERROR PC
864	003266	003322			ERRPC		
865	003270	104401			TYPE		:TYPE REST OF HEADER
866	003272	011001			CONERR		
867	003274	012737	177777 001302	IS:	MOV	#-1,\$TMP2	:SET FLAG SO IT ONLY GETS TYPED ONCE
868	003302	010137	001264		MOV	R1,\$REG1	:SAVE R1 FOR TYPEOUT
869	003306	104416			CONVRT		
870	003310	003330			CONTAB		:TYPE CSR VALUES
871	003312	104401		3S:	TYPE		
872	003314	011022			KMCH		
873	003316	022626		4S:	CMP	(SP)+,(SP)+	:ADJUST STACK
874	003320	000737			BR	OK	:BR TO GET OUT
875	003322	000001		ERRPC:	1		
876	003324	006	002		.BYTE	6,2	
877	003326	001460			SAVPC		
878	003330	000002		CONTAB:	2		
879	003332	006	004		.BYTE	6,4	
880	003334	001264			\$REG1		
881	003336	006	002		.BYTE	6,2	
882	003340	002066			KMCSR		
883	003342	007		DEVTAB:	.BYTE	7	:DJ
884	003343	017			.BYTE	17	:DH
885	003344	007			.BYTE	7	:DQ
886	003345	007			.BYTE	7	:DU
887	003346	007			.BYTE	7	:DUP
888	003347	007			.BYTE	7	:LK
889	003350	007			.BYTE	7	:DMC
890	003351	007			.BYTE	7	:DZ
891	003352	007			.BYTE	7	:KMC


```

003354 003354 012637 000006 .EVEN
003354 012637 000004 AUDONE:
003360 032737 000010 001446 1S: MOV (SP)+,206 ;RESTORE LOC 6
003372 001422 BIT #503,STRTSW ;RESTORE LOC 4
003374 104401 007771 BEQ 3S ;SELECT SPECIFIC DEVICES??
003400 005000 CLR R0 ;BR IF NO.
003400 000000 TYPE MNEW ;TYPE THE MESSAGE.
003400 000000 CLR R0 ;ZERO DATA LIGHTS
003404 027737 175630 001474 HALT ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
003412 101404 CMP #SMR,SAVACT ;IS THE NUMBER VALID?
003414 104401 007644 BLOS 2S ;BR IF NUMBER IS OK.
003420 000000 TYPE ,MERR3 ;TELL USER OF INVALID NUMBER.
003424 000776 HALT ;STOP EVERY THING.
003432 017737 175610 001470 2S: BR -2 ;RESTART THE PROGRAM AGAIN.
003436 013700 001470 MOV #SMR,KMACTV ;GET NEW DEVICE PATTERN
003440 012700 000300 HALT KMACTV,R0 ;SHOW THE USER WHAT HE SELECTED.
003444 012701 000302 3S: MOV #300,R0 ;CONTINUE DYNAMIC SWITCHES.
003450 010120 4S: MOV #302,R1 ;PREPARE TO CLEAR THE FLOATING
003452 005021 CLR (R1)+ ;VECTOR AREA. 300-776
003454 022021 CMP (R0)+,(R1)+ ;START PUTTING "PC+2 - HALT"
003456 022700 001000 CMP #1000,R0 ;IN VECTOR AREA.
003462 001372 BNE 4S ;POP POINTERS
;ALL DONE??
;BR IF NO.

;TEST START AND RESTART
-----
003464 012706 001200 .BEGIN: MOV #STACK,SP ;SET UP STACK
003470 013746 000006 MOV 206,-(SP) ;SAVE LOC 6
003474 013746 000004 MOV 204,-(SP) ;SAVE LOC 4
003500 005000 CLR R0 ;START AT 0
003502 012737 003546 000004 6S: MOV #2S,204 ;SET UP FOR TIME OUT
003510 005037 000006 CLR 206 ;TO AUTOSIZE MEMORY
003514 005720 TST (R0)+ ;CHECK ADDRESS IN R0
003516 022700 157776 CMP #157776,R0 ;IS IT AT LEAST 28K
003518 001374 BNE 6S ;BR IF NO
003524 162700 007776 SUB #7776,R0 ;SAVE 2K FOR MONITORS
003530 010037 001466 7S: MOV R0,MEALIM ;STORE MEMORY LIMIT
003534 012637 000004 MOV (SP)+,204 ;RESTORE LOC 4
003540 012637 000006 MOV (SP)+,206 ;RESTORE LOC 6
003544 000413 BR 10S ;CONTINUE
003546 022626 2S: CMP (SP)+,(SP)+ ;ADJUST STACK
003550 162700 000004 SUB #4,R0 ;GET LAST GOOD ADDRESS
003554 162700 007776 SUB #7776,R0 ;SAVE 2K FOR MONITORS
003560 022700 030000 CMP #30000,R0 ;IS IT 8K?
003564 001361 BNE 7S ;BR IF NO
003566 012700 037400 MOV #37400,R0 ;IF 8K DON'T SAVE 2K
003572 000756 BR 7S
003574 012737 000340 177776 10S: MOV #340,PS ;LOCK OUT INTERRUPTS
003602 032737 000004 001446 BIT #BIT2,STRTSW ;CHECK FOR LOCK ON TEST
003610 001406 BEQ 1S ;BR IF NO LOCK DESIRED.
003612 104401 007670 TYPE MLOCK ;TYPE LOCK SELECTED.
003616 012737 000240 004146 MOV #NOP,TTST ;SET UP TO LOCK
003624 000403 BR 3S ;CONTINUE ALONG.
003626 013737 004354 004146 1S: MOV BRW,TTST ;PREPARE NORMAL SCOPE ROUTINE

```

PROGRAM INITIALIZATION AND START UP.

```

748 003634 012737 011464 001206 3$: MOV #CYCLE,SLPADR ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
749 003642 032737 000002 001446 4$: BIT #SW01,STRTSW ;IS TEST NO. SELECTED?
950 003650 001002 000000 000000 5$: BNE SS ;BR IF YES
951 003652 104401 007614 000000 6$: TYPE MR ;TYPE R
752 003656 000177 175324 000000 7$: JMP @SLPADR ;START TESTING

```

003662
003663
003664
003670
003674
003700
003704
003710
003714
003720
003724
003730
003734
003740
003744
003750
003754
003762
003766
003774
004000
004004
004006
004014
004022
004026
004032
004036
004044
004046
004050
004052
004054
004056
004060
004064
004066
004070
004072
004074
004076
004100
004100

000005
005237 001324
105037 001203
104401 007572
104401 007717
104417 004104
104401 007725
104417 004112
104401 007733
104417 004120
104401 007744
104417 004126
013700 001504
013720 001324
013720 001212
013777 002060 176074
005077 176072
013777 002064 176066
005077 176064
005337 001476
001035
112737 000377 001511
013737 001472 001476
005037 001216
005037 001310
005237 001324
042737 100000 001324
005327
000001
003013
012737
000001
004046
013700 000042
001405
000005
004710
000240
000240
000240
000137

```

:END OF PASS
:TYPE NAME OF TEST
:UPDATE PASS COUNT
:CHECK FOR EXIT TO ACT-11
:RESTART TEST

.SBTTL END OF PASS ROUTINE

:*****
:*INCREMENT THE PASS NUMBER (SPASS)
:*IF THERES A MONITOR GO TO IT
:*IF THERE ISN'T JUMP TO CYCLE

SEOP:
  RESET
  INC SPASS
  CLAB SERFLG
  TYPE ,NEPASS
  TYPE ,NCSRX
  CNVRT ,XCSR
  TYPE ,NVECX
  CNVRT ,XVEC
  TYPE ,NPASSX
  CNVRT ,XPASS
  TYPE ,NERRX
  CNVRT ,XERR
  MOV RILK,RO
  MOV SPASS,(RO)+
  MOV SENTTL,(RO)+
  MOV K0RLVL,20RVEC
  CLR 20RVL
  MOV K0TLVL,20TVEC
  CLR 20TTLVL
  DEC SAVNUM
  BNE SDOAGN
  MOVB 8377,QV.FLG
  MOV K0RNUM,SAVNUM
  CLR SERAPC
  CLR STINES
  INC SPASS
  BIC #10000,SPASS
  DEC (PC)+

  SEOPCT: .WORD 1
  BGT SDOAGN
  MOV (PC)+,2(PC)+

  SENDCT: .WORD 1
  SEOPCT

  SGET42: MOV 2#42,RO
  BEQ SDOAGN

  SENDAD: JSR PC,(RO)
  NOP
  NOP
  NOP

  SDOAGN: JMP 2(PC)+

  INCREMENT THE PASS COUNT
  CLEAR ERROR FLAG
  TYPE END PASS.
  TYPE "CSR"
  SHOW IT.
  TYPE VECTOR.
  SHOW IT.
  TYPE "PASSES -"
  SHOW IT.
  TYPE "ERRORS -"
  SHOW IT.
  SET POINTER TO PASSCNT.
  SAVE THE PASS COUNT.
  SAVE ERROR COUNT
  RESTORE THE RECEIVER INTERRUPT VECTOR.
  RESTORE RECEIVER LEVEL
  RESTORE THE TRANSMIT INTERRUPT VECTOR.
  RESTORE TRANSMITTER LEVEL
  ALL DEVICE TESTED?
  BRANCH IF NO.
  SET QUICK VERIFY FLAG.
  RESTORE DEVICE COUNT.
  CLEAR LAST ERROR PC
  ZERO THE NUMBER OF ITERATIONS
  INCREMENT THE PASS NUMBER
  DON'T ALLOW A NEG. NUMBER
  LOOP?

  ;; YES
  ;; RESTORE COUNTER

  ;; GET MONITOR ADDRESS
  ;; BRANCH IF NO MONITOR
  ;; CLEAR THE WORLD
  ;; GO TO MONITOR
  ;; SAVE ROOM
  ;; FOR
  ;; ACT11

  ;; RETURN
  
```

END OF PASS ROUTINE

1009	004102	011464	
1010	004104	000001	
1011	004106	006	002
1012	004110	002066	
1013	004112	000001	
1014	004114	004	002
1015	004116	002056	
1016	004120	000001	
1017	004122	006	002
1018	004124	001324	
1019	004126	000001	
1020	004130	006	002
1021	004132	001212	

```

SRTNAD: .WORD   CYCLE
XCSR:   1
        .BYTE   6,2
XVEC:   1
        .BYTE   4,2
XPASS:  1
        .BYTE   6,2
XERR:   1
        .BYTE   6,2
SERTTL

```

;SCOPE LOOP AND ITERATION HANDLER

.SBTTL SCOPE HANDLER ROUTINE

1022			
1023			
1024			
1025			
1026			
1027			
1028			
1029			
1030			
1031			
1032			
1033			
1034			
1035			
1036			
1037			
1038	004134		
1039	004134	005037	001216
1040	004140	023716	013776
1041	004144	001413	
1042	004146	000406	
1043	004150	105777	175070
1044	004154	100067	
1045	004156	017766	175064 177776
1046	004164	032777	040000 175046
1047	004172	001060	
1048			
1049	004174	000416	
1050			
1051	004176	013746	000004
1052	004202	012737	004222 000004
1053	004210	005737	177060
1054	004214	012637	000004
1055	004220	000436	
1056	004222	022626	
1057	004224	012637	000004
1058	004230	000441	
1059	004232		
1060	004232	105737	001203
1061	004236	001404	
1062	004240	105037	001203
1063	004244	005037	001310
1064	004250	032777	004000 174762

```

*****
;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
;AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
;AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
;SM14=1      LOOP ON TEST
;SM11=1      INHIBIT ITERATIONS
;CALL
;          SCOPE          ;;SCOPE=IOT
SSCOPE:   CLR          $ERRPC          ; CLEAR LAST ERROR PC
          CMP          TST1+2,(SP)      ; IS THIS TEST #1 ?
          BEQ          $XTSTR          ; IF SO DON'T LOOP.
TTST:     BR          IS
          TSTB         $STKS          ; KEYBOARD DONE ?
          BPL          $OVER          ; IF NO DONT WAIT.
          MOV          $STKB,-2(SP)
IS:       BIT          $BIT14,$SMR
          BNE          $OVER
;*****START OF CODE FOR THE XOR TESTER*****
$XTSTR:   BR          6S
          MOV          $NERRVEC,-(SP)
          MOV          $S,$NERRVEC
          TST         $#177060
          MOV          (SP)+,$NERRVEC
          BR          $SVLAD
5S:       CMP          (SP)+,(SP)+
          MOV          (SP)+,$NERRVEC
          BR          $OVER
6S:;*****END OF CODE FOR THE XOR TESTER*****
2S:       TSTB         $ERFLG
          BEQ          3S
4S:       CLR          $ERFLG
          CLR          $TIMES
          BIT          $BIT11,$SMR
          ; LOOP ON PRESENT TEST?
          ; YES IF SM14=1
          ; IF RUNNING ON THE "XOR" TESTER CHANGE
          ; THIS INSTRUCTION TO A "NOP" (NOP=240)
          ; SAVE THE CONTENTS OF THE ERROR VECTOR
          ; SET FOR TIMEOUT
          ; TIME OUT ON XOR?
          ; RESTORE THE ERROR VECTOR
          ; GO TO THE NEXT TEST
          ; CLEAR THE STACK AFTER A TIME OUT
          ; RESTORE THE ERROR VECTOR
          ; LOOP ON THE PRESENT TEST
          ; HAS AN ERROR OCCURRED?
          ; BR IF NO
          ; ZERO THE ERROR FLAG
          ; CLEAR THE NUMBER OF ITERATIONS TO MAKE
          ; INHIBIT ITERATIONS?

```

```

1065 004256 001011 BNE 15 ; BR IF YES
1066 004260 005737 001324 TST SPASS ; IF FIRST PASS OF PROGRAM
1067 004264 001406 BEQ 15 ; INHIBIT ITERATIONS
1068 004266 005237 001204 INC $ICNT ; INCREMENT ITERATION COUNT
1069 004272 023737 001310 001204 CMP $TIMES,$ICNT ; CHECK THE NUMBER OF ITERATIONS MADE
1070 004300 002015 BGE $OVER ; BR IF MORE ITERATION REQUIRED
1071 004302 012737 000001 001204 1S: MOV $I,$ICNT ; REINITIALIZE THE ITERATION COUNTER
1072 004310 013737 004356 001310 MOV $SMXCNT,$TIMES ; SET NUMBER OF ITERATIONS TO DO
1073 004316 105237 001202 $SVLAD: INCB $STNM ; COUNT TEST NUMBERS
1074 004322 113737 001202 001322 MOV $STNM,$STESTN ; SET TEST NUMBER IN APT MAILBOX
1075 004330 011637 001206 MOV (SP), $LPAOR ; SAVE SCOPE LOOP ADDRESS
1076 004334 013777 001202 174700 $OVER: MOV $STNM,$DISPLAY ; DISPLAY TEST NUMBER
1077 004342 013716 001206 MOV $LPAOR,(SP) ; FUDGE RETURN ADDRESS
1078 004346 013701 002066 MOV $KMSCR,$R1 ; R1 CONTAINS BASE KMC ADDRESS.
1079 004352 000002 RTI
1080 004354 000406 BRW: WORD 406
1081 004356 000020 $SMXCNT: 20 ; ;MAX. NUMBER OF ITERATIONS
1082
1083 ;CHECK FOR FREEZE ON CURRENT DATA
1084 -----
1085
1086 004360 004737 011216 .SCOPI: JSR PC,$KMSR ;CHECK FOR SOFT SWR
1087 004364 032777 001000 174646 BIT $SM09,$SWR ; IS SM09=1(SET)?
1088 004372 001403 BEQ 15 ; BR IF NOT SET.
1089 004374 012737 177777 001444 MOV $-1,$LOCK ; FLAG IT.
1090 004402 000002 1S: RTI ; GO BACK.
1091
1092 ;TELETYPE OUTPUT ROUTINE
1093 -----
1094
1095 .SBTTL TYPE ROUTINE
1096
1097 ;*****
1098 ;ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
1099 ;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
1100 ;NOTE1: $FILLC CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
1101 ;NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
1102 ;NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
1103 ;
1104 ;CALL:
1105 ;1) USING A TRAP INSTRUCTION
1106 ; TYPE ,MESADR ; ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
1107 ;OR
1108 ; TYPE
1109 ; MESADR
1110 ;
1111
1112 004404 105737 001257 $TYPE: TSTB $TFPLG ; IS THERE A TERMINAL?
1113 004410 100002 BPL 15 ; BR IF YES
1114 004412 000000 HALT ; HALT HERE IF NO TERMINAL
1115 004414 000430 BR 35 ; LEAVE
1116 004416 010046 1S: MOV $R0,-(SP) ; SAVE R0
1117 004420 017600 000002 MOV $R0,$R0 ; GET ADDRESS OF ASCIZ STRING
1118 004424 122737 000001 001336 CMPB $APTEVM,$ENV ; RUNNING IN APT MODE
1119 004432 001011 BNE 62$ ; NO, GO CHECK FOR APT CONSOLE
1120 004434 132737 000100 001337 BITB $APTSPOOL,$ENVM ; SPOOL MESSAGE TO APT

```

```

1121 004442 001405      BEQ      625      ;; NO, GO CHECK FOR CONSOLE
1122 004444 010037 004454  MOV      RD, 615  ;; SETUP MESSAGE ADDRESS FOR APT
1123 004450 004737 004674  JSR      PC, SATY3  ;; SPOOL MESSAGE TO APT
1124 004454 000000      JSR      0          ;; MESSAGE ADDRESS
1125 004456 132737 000040 001337 61S:    .WORD      0          ;; APT CONSOLE SUPPRESSED
1126 004458 001003      BNE      608      ;; YES, SKIP TYPE OUT
1127 004462 112046      MOV      (RD)+, -(SP)  ;; PUSH CHARACTER TO BE TYPED ONTO STACK
1128 004470 001005      BNE      45       ;; BR IF IT ISN'T THE TERMINATOR
1129 004472 006726      TST      (SP)+      ;; IF TERMINATOR POP IT OFF THE STACK
1130 004474 012500      MOV      (SP)+, RD   ;; RESTORE RD
1131 004476 002716 000002 3S:     ADD      #2, (SP)   ;; ADJUST RETURN PC
1132 004508 000002      RTI          ;; RETURN
1133 004504 122716 000011 4S:     CNPB    #HT, (SP)  ;; BRANCH IF <HT>
1134 004510 001430      BEQ      85       ;;
1135 004512 122716 000200  CNPB    #CRLF, (SP)  ;; BRANCH IF NOT <CRLF>
1136 004516 001006      BNE      55       ;;
1137 004520 006726      TST      (SP)+      ;; POP <CR><LF> EQUIV
1138 004522 104401      TYPE    ;; TYPE A CR AND LF
1139 004524 001313      SCRLF    ;;
1140 004526 105037 004662  CLRB    #SCHARCNT  ;; CLEAR CHARACTER COUNT
1141 004530 000735      BR      25       ;; GET NEXT CHARACTER
1142 004534 004737 004616 5S:     JSR      PC, STYPEC  ;; GO TYPE THIS CHARACTER
1143 004540 123726 001256 6S:     CNPB    #FILLC, (SP)+  ;; IS IT TIME FOR FILLER CHARS.?
1144 004542 001350      BNE      65       ;; IF NO GO GET NEXT CHAR.
1145 004546 013746 001254  MOV      #NULL, -(SP)  ;; GET # OF FILLER CHARS. NEEDED
1146                                     ;; AND THE NULL CHAR.
1147 004552 105366 000001 7S:     DECB    1, (SP)    ;; DOES A NULL NEED TO BE TYPED?
1148 004556 002770      BLT     65       ;; BR IF NO--GO POP THE NULL OFF OF STACK
1149 004560 004737 004616  JSR      PC, STYPEC  ;; GO TYPE A NULL
1150 004564 105337 004662  DECB    #SCHARCNT  ;; DO NOT COUNT AS A COUNT
1151 004570 000770      BR      75       ;; LOOP
1152
1153 ;HORIZONTAL TAB PROCESSOR
1154
1155 004572 112716 000040 8S:     MOV      #', (SP)  ;; REPLACE TAB WITH SPACE
1156 004576 004737 004616 9S:     JSR      PC, STYPEC  ;; TYPE A SPACE
1157 004602 132737 000007 004662 BITB    #7, #SCHARCNT  ;; BRANCH IF NOT AT
1158 004610 001372      BNE      95       ;; TAB STOP
1159 004612 005726      TST      (SP)+      ;; POP SPACE OFF STACK
1160 004614 000724      BR      25       ;; GET NEXT CHARACTER
1161 004616 105777 174426 STYPEC: TSTB    #STPS  ;; WAIT UNTIL PRINTER IS READY
1162 004622 100375      BPL     STYPEC    ;;
1163 004624 116677 000002 174420 MOV      2(SP), #STPB  ;; LOAD CHAR TO BE TYPED INTO DATA REG.
1164 004632 122766 000015 000002 CNPB    #CR, 2(SP)  ;; IS CHARACTER A CARRIAGE RETURN?
1165 004640 001003      BNE      15       ;; BRANCH IF NO
1166 004642 105037 004662  CLRB    #SCHARCNT  ;; YES--CLEAR CHARACTER COUNT
1167 004646 000406      BR      STYPEX    ;; EXIT
1168 004650 122766 000012 000002 1S:     CNPB    #LF, 2(SP)  ;; IS CHARACTER A LINE FEED?
1169 004656 001402      BEQ     STYPEX    ;; BRANCH IF YES
1170 004660 105227      INCB    (PC)+      ;; COUNT THE CHARACTER
1171 004662 000000      SCHARCNT: .WORD 0  ;; CHARACTER COUNT STORAGE
1172 004664 000207      STYPEX: RTS      PC
1173
1174 .SBTTL  APT COMMUNICATIONS ROUTINE
1175
1176 ;*****

```


APT COMMUNICATIONS ROUTINE

```

1177 004666 112737 000001 005132 SATY1: MOVB #1, SFFLG ;; TO REPORT FATAL ERROR
1178 004674 112737 000001 005130 SATY3: MOVB #1, SFFLG ;; TO TYPE A MESSAGE
1179 004702 000403 BR SATYC
1180 004704 112737 000001 005132 SATY4: MOVB #1, SFFLG ;; TO ONLY REPORT FATAL ERROR
1181 004712 SATYC:
1182 004712 010046 MOV RO, -(SP) ;; PUSH RO ON STACK
1183 004714 010146 MOV R1, -(SP) ;; PUSH R1 ON STACK
1184 004716 105737 005130 TSTB SFFLG ;; SHOULD TYPE A MESSAGE?
1185 004722 001450 BEQ 55 ;; IF NOT: BR
1186 004724 122737 000001 001336 CMPB #APTENV, SENV ;; OPERATING UNDER APT?
1187 004732 001031 BNE 35 ;; IF NOT: BR
1188 004734 132737 000100 001337 BITB #APTPOOL, SENVM ;; SHOULD SPOOL MESSAGES?
1189 004742 001425 BEQ 35 ;; IF NOT: BR
1190 004744 017600 000004 MOV #4(SP), RO ;; GET MESSAGE ADDR.
1191 004750 062766 000002 000004 ADD #2, 4(SP) ;; BUMP RETURN ADDR.
1192 004756 005737 001316 15: TST SMSGTYPE ;; SEE IF DONE W/ LAST XMISSION?
1193 004762 001375 BNE 15 ;; IF NOT: WAIT
1194 004764 010037 001332 MOV RO, SMSGADR ;; PUT ADDR IN MAILBOX
1195 004770 105720 25: TSTB (RO)+ ;; FIND END OF MESSAGE
1196 004772 001376 BNE 25
1197 004774 163700 001332 SUB SMSGADR, RO ;; SUB START OF MESSAGE
1198 005000 006200 ASR RO ;; GET MESSAGE LGTH IN WORDS
1199 005002 010037 001334 MOV RO, SMSGLGT ;; PUT LENGTH IN MAILBOX
1200 005006 012737 000004 001316 MOV #4, SMSGTYPE ;; TELL APT TO TAKE MSG.
1201 005014 000413 BR 55
1202 005016 017637 000004 005042 35: MOV #4(SP), 45 ;; PUT MSG ADDR IN JSR LINKAGE
1203 005024 062766 000002 000004 ADD #2, 4(SP) ;; BUMP RETURN ADDRESS
1204 005032 013746 177776 MOV 177776, -(SP) ;; PUSH 177776 ON STACK
1205 005036 004737 004404 JSR PC, $TYPE ;; CALL TYPE MACRO
1206 005042 000000 45: .WORD 0
1207 005044 55:
1208 005044 105737 005132 105: TSTB SFFLG ;; SHOULD REPORT FATAL ERROR?
1209 005060 001416 BEQ 125 ;; IF NOT: BR
1210 005062 005737 001336 TST SENV ;; RUNNING UNDER APT?
1211 005066 001413 BEQ 125 ;; IF NOT: BR
1212 005068 005737 001316 115: TST SMSGTYPE ;; FINISHED LAST MESSAGE?
1213 005064 001375 BNE 115 ;; IF NOT: WAIT
1214 005066 017637 000004 001320 MOV #4(SP), SFATAL ;; GET ERROR #
1215 005074 062766 000002 000004 ADD #2, 4(SP) ;; BUMP RETURN ADDR.
1216 005102 005237 001316 INC SMSGTYPE ;; TELL APT TO TAKE ERROR
1217 005106 105037 005132 125: CLRB SFFLG ;; CLEAR FATAL FLAG
1218 005112 105037 005131 CLRB SLFLG ;; CLEAR LOG FLAG
1219 005116 105037 005130 CLRB SMFLG ;; CLEAR MESSAGE FLAG
1220 005122 012601 MOV (SP)+, R1 ;; POP STACK INTO R1
1221 005124 012600 MOV (SP)+, RO ;; POP STACK INTO RO
1222 005126 000207 RTS ;; RETURN
1223 005130 000 .BYE PC
1224 005131 000 .BYE 0
1225 005132 000 .BYE 0
1226 005134 .EVEN
1227 000200 APTSIZE=200
1228 000001 APTENV=001
1229 000100 APTPOOL=100
1230 000040 APTCSUP=040
1231
1232

```

TTY INPUT ROUTINE

.SBTTL TTY INPUT ROUTINE

:::*****

.ENABL LSB

.DSABL LSB

:::*****

THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY

#CALL:

* R0CHR
* RETURN HERE

::: INPUT A SINGLE CHARACTER FROM THE TTY
::: CHARACTER IS ON THE STACK
::: WITH PARITY BIT STRIPPED OFF

005134 011646
005135 016666 000004 000002
005136 105777 174074
005137 100375
005138 117766 174070 000004
005139 042766 177600 000004
005140 022627 000004 000023
005141 001013
005142 105777 174042
005143 100375
005144 117766 174036
005145 042766 177600
005146 022627 000021
005147 001366
005148 000750
005149 022627 000004 000140
005150 002407
005151 022627 000004 000175
005152 003003
005153 042766 000040 000004
005154 000002

SROCHR: MOV (SP), -(SP)
MOV 4(SP), 2(SP)
15: TSTB 25TKS
BPL 15
MOV 25TKB, 4(SP)
BIC #1C(177), 4(SP)
CMP 4(SP), #23
BNE 35
25: TSTB 25TKS
BPL 25
MOV 25TKB, -(SP)
BIC #1C(177), (SP)
CMP (SP)+, #21
BNE 25
BR 15
35: CMP 4(SP), #140
BLT 45
CMP 4(SP), #175
BGT 45
45: BIC #40, 4(SP)
RTI

::: PUSH DOWN THE PC
::: SAVE THE PS
::: WAIT FOR
::: A CHARACTER
::: READ THE TTY
::: GET RID OF JUNK IF ANY
::: IS IT A CONTROL-S?
::: BRANCH IF NO
::: WAIT FOR A CHARACTER
::: LOOP UNTIL ITS THERE
::: GET CHARACTER
::: MAKE IT 7-BIT ASCII
::: IS IT A CONTROL-0?
::: IF NOT DISCARD IT
::: YES, RESUME
::: IS IT UPPER CASE?
::: BRANCH IF YES
::: IS IT A SPECIAL CHAR?
::: BRANCH IF YES
::: MAKE IT UPPER CASE
::: GO BACK TO USER

:::*****

THIS ROUTINE WILL INPUT A STRING FROM THE TTY

#CALL:

* RDLIN
* RETURN HERE

::: INPUT A STRING FROM THE TTY
::: ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
::: TERMINATOR WILL BE A BYTE OF ALL 0'S

005251 010346
005252 005046
005253 012703 005510
005254 022703 005517
005255 101456
005256 104402
005257 112613
005258 122713 000177
005259 001022
005260 006716
005261 001007
005262 112737 000134 005506

SROLIN: MOV R3, -(SP)
CLR -(SP)
15: MOV #TTYIN, R3
25: CMP #TTYIN+7, R3
BLOS 45
R0CHR
MOV (SP)+, (R3)
105: CMP #177, (R3)
BNE 55
TST (SP)
BNE 65
MOV #'\, 95

::: SAVE R3
::: CLEAR THE RUBOUT KEY
::: GET ADDRESS
::: BUFFER FULL?
::: BR IF YES
::: GO READ ONE CHARACTER FROM THE TTY
::: GET CHARACTER
::: IS IT A RUBOUT
::: BR IF NO
::: IS THIS THE FIRST RUBOUT?
::: BR IF NO
::: TYPE A BACK SLASH

1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288

1330	005506	104401	005506	TYPE	95		
1331	005506	012716	177777	MOV	1-1, (SP)		SET THE RUBOUT KEY
1332	005506	005506		DEC	R3		BACKUP BY ONE
1333	005506	005506		6S: CHP	R3, #STTYIN		STACK EMPTY?
1334	005506	005510		BLO	45		BR IF YES
1335	005506	111337	005506	NOVB	(R3), 95		SETUP TO TYPEOUT THE DELETED CHAR.
1336	005506	104401	005506	TYPE	95		GO TYPE
1337	005506	000746		BR	25		GO READ ANOTHER CHAR.
1338	005506	005716		5S: TST	(SP)		RUBOUT KEY SET?
1339	005506	001406		BEG	75		BR IF NO
1340	005506	112737	000134 005506	NOVB	0', 95		TYPE A BACK SLASH
1341	005506	104401	005506	TYPE	95		
1342	005506	005016		CLR	(SP)		CLEAR THE RUBOUT KEY
1343	005506	122713	000025	7S: CHPB	825, (R3)		IS CHARACTER A CTRL U?
1344	005506	001003		BNE	85		BR IF NO
1345	005506	104401	005517	TYPE	, SCNTLU		TYPE A CONTROL "U"
1346	005506	000726		BR	15		GO START OVER
1347	005506	122713	000022	8S: CHPB	822, (R3)		IS CHARACTER A "1R"?
1348	005506	001011		BNE	35		BRANCH IF NO
1349	005506	105013		CLRB	(R3)		CLEAR THE CHARACTER
1350	005506	104401	001313	TYPE	, SCRLF		TYPE A "CR" & "LF"
1351	005506	104401	005510	TYPE	, STTYIN		TYPE THE INPUT STRING
1352	005506	000717		BR	25		GO PICKUP ANOTHER CHARACTER
1353	005506	104401	001312	4S: TYPE	, SQUES		TYPE A '?'
1354	005506	000712		BR	15		CLEAR THE BUFFER AND LOOP
1355	005506	111337	005506	3S: NOVB	(R3), 95		ECHO THE CHARACTER
1356	005506	104401	005506	TYPE	95		
1357	005506	122723	000015	CHPB	815, (R3)+		CHECK FOR RETURN
1358	005506	001305		BNE	25		LOOP IF NOT RETURN
1359	005506	105063	177777	CLRB	-1(R3)		CLEAR RETURN (THE 15)
1360	005506	104401	001314	TYPE	, SLF		TYPE A LINE FEED
1361	005506	005769		TST	(SP)+		CLEAR RUBOUT KEY FROM THE STACK
1362	005506	012603		NOV	(SP)+, R3		RESTORE R3
1363	005506	011646		NOV	(SP) - (SP)		ADJUST THE STACK AND PUT ADDRESS OF THE
1364	005506	016666	000004 000002	NOV	4(SP), 2(SP)		FIRST ASCII CHARACTER ON IT
1365	005506	012766	005510 000004	NOV	#STTYIN, 4(SP)		
1366	005506	000002		RTI			RETURN
1367	005506	000		9S: .BYTE	0		STORAGE FOR ASCII CHAR. TO TYPE
1368	005506	000		.BYTE	0		TERMINATOR
1369	005506	000007		STTYIN: .BLKB	7		RESERVE 7 BYTES FOR TTY INPUT
1370	005506	005525 000012		SCNTLU: .ASCIZ	/'U/'(15)'(12)		CONTROL "U"
1371	005506	043536 000		SCNTLG: .ASCIZ	/'G/'(15)'(12)		CONTROL "G"
1372	005506	051412 051127		SMSR: .ASCIZ	'(15)'(12)/SMR = /		
1373	005506	036440 000040		SMNEW: .ASCIZ	/'NEW = /		
1374	005506	020040 042516	020127				
1375	005506	020075 000					
1376	005506	005554					

```

.EVEN
.SBTTL READ AN OCTAL NUMBER FROM THE TTY

*****
*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
*CHANGE IT TO BINARY.
*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.

```

1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380

READ AN OCTAL NUMBER FROM THE TTY

```

1345      ;*CALL:
1346      ;*      RDOCT
1347      ;*      RETURN HERE
1348      ;*
1349      ;*      READ AN OCTAL NUMBER
1350      ;*      LOW ORDER BITS ARE ON TOP OF THE STACK
1351      ;*      HIGH ORDER BITS ARE IN SHIOCT
1352
1353      005554  011646  000004  000002  SRDOCT:  MOV      (SP),-(SP)
1354      005555  011646  000004  000002  4(SP),R5
1355      005556  010046  000004  000002  MOV      R0,-(R5)
1356      005557  010146  000004  000002  MOV      R1,-(R5)
1357      005558  010246  000004  000002  MOV      R2,-(R5)
1358      005559  104403  000004  000002  1S:     ROL IN
1359      005560  012500  005702  000002  MOV      (SP)+,R0
1360      005561  010007  005702  000002  MOV      R0,R5
1361      005562  005001  005702  000002  CLR      R1
1362      005563  005002  005702  000002  CLR      R2
1363      005564  112046  005702  000002  2S:     MOVB   (R0)+,-(SP)
1364      005565  001420  000060  000002  BEQ      3S
1365      005566  122716  000060  000002  CNPB   #'0,(SP)
1366      005567  003026  000067  000002  BGT     4S
1367      005568  122716  000067  000002  CNPB   #'7,(SP)
1368      005569  003423  000067  000002  BLT     4S
1369      005570  006301  000067  000002  ROL     R1
1370      005571  006102  000067  000002  ROL     R2
1371      005572  006301  000067  000002  ROL     R1
1372      005573  006102  000067  000002  ROL     R2
1373      005574  042716  177770  000002  BIC     #'C7,(SP)
1374      005575  062601  000002  000002  ROR     (SP)+,R1
1375      005576  000756  000002  000002  BR      2S
1376      005577  010166  000012  000002  3S:     TST     (SP)+
1377      005578  010237  005712  000002  MOV     R1,12(SP)
1378      005579  012602  005712  000002  MOV     R2,SHIOCT
1379      005580  012601  005712  000002  MOV     (SP)+,R2
1380      005581  012600  005712  000002  MOV     (SP)+,R1
1381      005582  000002  005712  000002  MOV     (SP)+,R0
1382      005583  005726  005712  000002  RTI
1383      005584  105010  005712  000002  4S:     TST     (SP)+
1384      005585  104401  005712  000002  CLRB   (R0)
1385      005586  000000  005712  000002  TYPE
1386      005587  104401  001312  000002  5S:     .WORD  0
1387      005588  000730  001312  000002  TYPE   'QUES
1388      005589  000000  001312  000002  BR      1S
1389      SHIOCT: .WORD  0
1390
1391      INPUT OCTAL NUMBER ROUTINE
1392
1393      005714  010546  000002  000002  $INPUT: MOV     R5,-(SP)
1394      005715  016605  000002  000002  MOV     2(SP),R5
1395      005716  012537  005760  000002  MOV     (R5)+,WHAT
1396      005717  012537  005760  000002  MOV     (R5)+,LOLIM
1397      005718  012537  005760  000002  MOV     (R5)+,HILIM
1398      005719  012537  005760  000002  MOV     (R5)+,WHERE
1399      005720  112537  005760  000002  MOVB   (R5)+,LOBITS
1400      005721  112537  005760  000002  MOVB   (R5)+,ADRCNT
1401
1402      ;*      SAVE REGISTER R5.
1403      ;*      GET FIRST PARAMETER ADDRESS.
1404      ;*      GET MESSAGE ADDRESS.
1405      ;*      GET LOW LIMIT FOR THE #.
1406      ;*      GET HIGH LIMIT FOR THE #.
1407      ;*      GET ADDRESS OF INBUFFER.
1408      ;*      GET LOWMASK BITS.
1409      ;*      GET # OF #'S TO BE GENERATED.

```

READ AN OCTAL NUMBER FROM THE TTY

140 005752 010566 000002
 141 005753 104401
 142 005754 000000
 143 005755 104404
 144 005756 021637 006042
 145 005757 003003
 146 005758 021637 006040
 147 005759 002005
 148 005760 104401 001312
 149 005761 104401 001313
 150 005762 000762
 151 005763 013705 006047
 152 005764 011625
 153 005765 022716 006047
 154 005766 105337
 155 005767 001372
 156 005768 005726
 157 005769 012605
 158 005770 000002
 159 005771 000000
 160 005772 000000
 161 005773 000000
 162 005774 000
 163 005775 000

INLP1: MOV RS,2(SP)
 WHAT: TYPE (SP),HILIM
 .WORD 0
 RDOCT (SP),LOLIM
 CNP 25
 BGT 35
 CNP 45
 BGE 55
 25: TYPE .SOLES
 TYPE .SCALE
 BR INLP1
 35: MOV WHERE,RS
 45: MOV (SP),(RS)+
 ADD R2,(SP)
 DECB ADRCNT
 BNE 45
 TST (SP)+
 MOV (SP)+,RS
 RTI
 LOLIM: .WORD 0
 HILIM: .WORD 0
 WHERE: .WORD 0
 LOBITS: .BYTE 0
 ADRCNT: .BYTE 0

SAVE THE RETURN ADDRESS.
 TYPE THE MESSAGE.
 READ OCTAL # FROM KEYBOARD.
 IS IT IN HIGH LIMIT?
 BRANCH IF NO.
 IS IT MORE THAN LOW LIMIT.
 BRANCH IF YES.
 TYPE " ? "
 TYPE <CR>,<LF>
 GET BUFFER ADDRESS.
 SAVE THE # IN RIGHT PLACE.
 NEXT SEQUENTIAL NUMBER.
 COUNT BY 1.
 BRANCH IF NOT DONE.
 POP THE STACK POINTER.
 POP THE REG.5

ADVANCE TO NEXT TEST HANDLER

142388 006050 013716 001442
 142389 006054 005037 001444
 142390 006060 000002
 142391
 142392
 142393
 142394
 142395 006062 016637 000004 001460
 142396
 142397
 142398
 142399 006070 010537 001274
 142400 006074 010437 001272
 142401 006108 010337 001270
 142402 006104 010237 001266
 142403 006110 010137 001264
 142404 006114 010037 001262
 142405 006120 000002
 142406
 142407
 142408
 142409 006122 013700 001262
 142410 006126 013701 001264
 142411 006130 013702 001266
 142412 006134 013703 001270
 142413 006138 013704 001272
 142414 006142 013705 001274
 142415 006152 000002

.ADVANCE: MOV NEXT,(SP)
 CLR LOCK
 RTI
 .SAV05: MOV 4(SP),SAVPC ;SAVE R7 (PC)
 ;SAVE R0-R5
 SV05: MOV R5,SREG5 ;SAVE R5
 MOV R4,SREG4 ;SAVE R4
 MOV R3,SREG3 ;SAVE R3
 MOV R2,SREG2 ;SAVE R2
 MOV R1,SREG1 ;SAVE R1
 MOV R0,SREG0 ;SAVE R0
 RTI ;LEAVE.
 ;RESTORE R0-R5
 .RES05: MOV SREG0,R0 ;RESTORE R0
 MOV SREG1,R1 ;RESTORE R1
 MOV SREG2,R2 ;RESTORE R2
 MOV SREG3,R3 ;RESTORE R3
 MOV SREG4,R4 ;RESTORE R4
 MOV SREG5,R5 ;RESTORE R5
 RTI ;LEAVE

CRUNCH STACK WITH ADDRESSOF SCOPE CALL
 RESET TIGHT LOOP ADDRESS
 CHECK TO SEE IF OLD TEST GETS REPEATED

CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER

1457					
1458					
1459					
1460	006154	104401	001313		
1461	006160	010046			
1462	006166	010146			
1463	006172	010346			
1464	006178	010446			
1465	006184	010546			
1466	006190	017601	000012		
1467	006196	062746	000002	000012	
1468	006202	012137	006376		
1470	006210	112137	006400		15:
1471	006216	112137	006401		
1472	006222	013137	006402		
1473	006228	122737	000003	006400	
1474	006234	001003			
1475	006240	042737	177400	006402	
1476	006246	013704	006402		25:
1477	006252	113705	006400		
1478	006258	012700	011112		
1479	006264	010403			35:
1480	006270	042703	177770		
1481	006276	062703	000060		
1482	006282	110320			
1483	006288	000041			
1484	006294	000041			
1485	006300	000041			
1486	006306	000041			
1487	006312	000041			
1488	006318	000041			
1489	006324	000041			
1490	006330	000041			
1491	006336	000041			
1492	006342	000041			
1493	006348	000041			
1494	006354	000041			
1495	006360	000041			
1496	006366	000041			
1497	006372	000041			
1498	006378	000041			
1499	006384	000041			
1500	006390	000041			
1501	006396	000041			
1502	006402	000041			
1503	006408	000041			
1504	006414	000041			
1505	006420	000041			
1506	006426	000041			
1507	006432	000041			
1508	006438	000041			
1509	006444	000041			
1510	006450	000041			
1511	006456	000041			
1512	006462	000041			

```

CONVR: TYPE SCALF
CNVRT: MOV R0, -(SP)
MOV R1, -(SP)
MOV R2, -(SP)
MOV R3, -(SP)
MOV R4, -(SP)
MOV R5, -(SP)
MOV #12(SP), R1
ADD #12(SP)
MOV (R1)+, WROCNT
MOV (R1)+, CHRCNT
MOV (R1)+, SPACNT
MOV #2(R1)+, BINMRD
CMPB R3, CHRCNT
BNE R5
BIC #177400, BINMRD
MOV BINMRD, R4
MOV CHRCNT, R5
MOV #TEMP, R0
MOV R4, R3
BIC #177770, R3
ADD #060, R3
MOV R3, (R0)+
CLC
ROR R4
CLC
ROR R4
CLC
ROR R4
DEC WROCNT
BNE WROCNT
MOV #INDATA, R3
MOV -(R0), (R3)+
DECB CHRCNT
BNE #5
TSTB SPACNT
BEQ #5
MOV #040, (R3)+
DECB SPACNT
BNE #5
CLRB (R3)
TYPE #INDATA
DEC WROCNT
BNE #15
MOV (SP)+, R5
MOV (SP)+, R4
MOV (SP)+, R3
MOV (SP)+, R1
MOV (SP)+, R0
RTI
WROCNT: 0
CHRCNT: 0
SPACNT=CHRCNT+1

```


1513 006402 000000

BINWRD: 0

1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568

006404 010046
006406 016600 000002
006412 005740
006414 111000
006416 006300
006420 016000 006440
006424 000200

; TRAP DISPATCH SERVICE
; ARGUMENT OF TRAP IS EXTRACTED
; AND USED AS OFFSET TO OBTAIN POINTER
; TO SELECTED SUBROUTINE

.SBTTL TRAP DECODER

; THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
; AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
; OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
; GO TO THAT ROUTINE.

STRAP: MOV RO, -(SP) ; SAVE RO
MOV 2(SP), RO ; GET TRAP ADDRESS
TST -(RO) ; BACKUP BY 2
MOVB (RO), RO ; GET RIGHT BYTE OF TRAP
ASL RO ; POSITION FOR INDEXING
MOV STRPAD(RO), RO ; INDEX TO TABLE
RTS RO ; GO TO ROUTINE

;; THIS IS USE TO HANDLE THE "GETPRI" MACRO

006426 011646
006430 016666 000004 000002
006436 000002

STRAP2: MOV (SP), -(SP) ; MOVE THE PC DOWN
MOV 4(SP), 2(SP) ; MOVE THE PSW DOWN
RTI ; RESTORE THE PSW

.SBTTL TRAP TABLE

; THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
; BY THE "TRAP" INSTRUCTION.

ROUTINE
STRPAD: .WORD STRAP2
STYPE ; ; CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
SROCHR ; ; CALL=ROCHR TRAP+2(104402) TTY TYPEIN CHARACTER ROUTINE
SROLIN ; ; CALL=ROLIN TRAP+3(104403) TTY TYPEIN STRING ROUTINE
SROOCT ; ; CALL=ROOCT TRAP+4(104404) READ AN OCTAL NUMBER FROM TTY
.SCOPI ; ; CALL=SCOPI TRAP+5(104405) CALL TO LOOP ON CURRENT DATA HANDLER
.SAVOS ; ; CALL=SAVOS TRAP+6(104406) CALL TO REGISTER SAVE ROUTINE
.RESOS ; ; CALL=RESOS TRAP+7(104407) CALL TO REGISTER RESTORE ROUTINE
.RSTCLR ; ; CALL=RSTCLR TRAP+10(104410) CALL TO ISSUE A MASTER CLEAR
.DELAY ; ; CALL=DELAY TRAP+11(104411) CALL TO DELAY
.RONCLK ; ; CALL=RONCLK TRAP+12(104412) CALL TO CLOCK ROM ONCE
.DATACLK ; ; CALL=DATACLK TRAP+13(104413) CALL TO CLOCK DATA
.TIMER ; ; CALL=TIMER TRAP+14(104414) CALL TO DELAY A CLOCK TICK
SINPUT ; ; CALL=INPUT TRAP+15(104415) CALL TO OCTAL # INPUT ROUTINE
.CONVRT ; ; CALL=CONVRT TRAP+16(104416) CALL TO
.CNVRT ; ; CALL=CNVRT TRAP+17(104417) CALL TO

```

1569 006500 006050      .ADVANCE      ;;CALL=ADVANCE TRAP+20(104420) CALL TO ADVANCE TO NEXT TEST
1570
1571 -----
1572 :*****
1573 :ERROR HANDLER
1574 :-----
1575
1576 006502 004737 011216 172524  ERROR:  JSR      PC,CKSMR      ;CHECK FOR SOFT SMR
1577 006506 032777 010000 172524  BIT      $SM12,$SMR    ;BELL ON ERROR?
1578 006514 001406 172526  BEQ      XBX          ;BR IF NO BELL
1579 006516 105777 172526  TSTB    $STPS        ;TTY READY
1580 006524 100003 000207 172520  BPL     XBX          ;DON'T WAIT IF TTY NOT READY.
1581 006524 112777 020000 172500  MOVSB   $207,$STPB   ;PUSH A BELL AT THE TTY.
1582 006524 032777 020000 172500  XBX:    BIT      $SM13,$SMR ;DELETE ERROR PRINT OUT?
1583 006524 001107 001216  BNE     HALTS        ;BR IF NO PRINT OUT WANTED.
1584 006524 021637 001216  CMP     (SP),SERAPC ;HAS THIS ERROR FOUND LAST TIME?
1585 006524 001404 001216  BEQ     IS           ;BR IF YES
1586 006524 011637 001216  MOV     (SP),SERAPC ;RECORD BEING HERE
1587 006524 105037 001203  CLRB   SERFLG       ;PREPARE HEADER
1588 006524 104406 1S:     SAVDS          ;SAVE ALL PROC REGISTERS
1589 006524 011E05 000002  MOV     (SP),R5     ;GET THE PC OF ERROR
1590 006524 162705 000002  SUB     $2,R5       ;GET ADDRESS OF TRAP CALL
1591 006570 011504 001214  MOV     (R5),R4    ;GET ERROR INSTRUCTION
1592 006572 110437 001214  MOVSB  R4,$ITEMB   ;COPY ERROR # FOR APT HANDLING
1593 006576 006304  ASL     R4          ;MULT BY TWO
1594 006600 061504  ADD     (R5),R4    ;DOUBLE IT
1595 006602 006304  ASL     R4          ;MULT AGAIN
1596 006604 042704 177001  BIC     $177001,R4 ;CLEAR JUNK
1597 006610 062704 001512  ADD     $SERATB,R4 ;GET POINTER
1598 006614 012437 006730  MOV     (R4)+,ERRMSG ;GET ERROR MESSAGE
1599 006620 012437 006742  MOV     (R4)+,DATAHD ;GET DATA HEADER
1600 006624 011437 006754  MOV     (R4),DATABP ;GET DATA TABLE
1601 006630 105737 001203  TSTB   SERFLG     ;TYPE HEADREER
1602 006634 001403  BEQ     TYPMSG     ;BR IF YES
1603 006636 005737 006754  TST    DATABP     ;DOES DATA TABLE EXIST?
1604 006642 001040  BNE     TYPDAT     ;BR IF YES.
1605 006644 104401 001313  TYPMSG: TYPE ,SCRLF ;
1606 006650 104401 001313  TYPE ,SCRLF ;
1607 006654 005737 001444  TST    LOCK       ;
1608 006660 001402  BEQ     IS         ;
1609 006662 104401 007767  TYPE ,MASTEK     ;
1610 006666 104401 007755 1S:    TYPE ,MTSTN     ;
1611 006672 104417 007072  CNVRT ,XTSTN     ;
1612 006676 104401 010044  TYPE ,MERRPC     ;
1613 006702 104417 007064  CNVRT ,ERTABD     ;
1614 006706 104401 001313  TYPE ,SCRLF     ;
1615 006712 112737 177777 001203  MOVSB  $-1,SERFLG ;
1616 006720 005737 006730  TST    ERRMSG     ;
1617 006724 001402  BEQ     WRKO.FM   ;
1618 006726 104401  TYPE          ;
1619 006730 000000  ERRMSG: 0        ;
1620 006732  WRKO.FM: 0        ;
1621 006732 005737 006742  TST    DATAHD   ;
1622 006736 001402  BEQ     TYPDAT   ;
1623 006740 104401  TYPE          ;
1624 006742 000000  DATAHD: 0      ;

```

```

1625 006744 005737 006754  TYPDAT: TST      DATARP      : DATA TABLE?
1626 006750 001402                BEQ      RESREG      : BR IF NO.
1627 006752 104416                CONVRT                : SHOW
1628 006754 000000                DATARP: 0              : DATA TABLE
1629 006756 104407                RESREG: RESOS         : RESTORE PROC REGISTERS
1630 006760 122737 000001 001336 HALTS:  CMPB      #APTEMV,SENV  : IS APT RUNNING?
1631 006762 001007                BNE                    : SKIP APT CALL IF NOT.
1632 006770 113737 001214 007002     MOVB      $ITEMB,6S   : COPY ERROR #.
1633 006776 004737 004704                JSR                    : CALL APT SERVICES.
1634 007000 000000                .WORD      0          : ERROR # GOES HERE.
1635 007004 000777                BR          9S        : LOCK HERE.
1636 007006 022737 004070 000042     CMP      #SENDAD,2#42 : IF ACT-11 AUTOMATIC MODE, HALT!!
1637 007014 001403                BEQ      1S          :
1638 007016 005777 172216                TST      2SMR        : HALT ON ERROR?
1639 007020 100005                BPL      EXITER      : BR IF NO HALT ON ERROR
1640 007024 010046                PUSHRO                : SAVE RO
1641 007026 016600 000002     MOV      2(SP),RO     : SHOW ERROR PC IN DATA LIGHTS
1642 007028 000000                HALT                  : HALT
1643 007034 012500                POPRO                 : GET RO
1644 007036 005237 001212     EXITER: INC      $ERTTL  : UPDATE ERROR COUNT
1645 007040 032777 000400 172170     BIT      #SM08,2SMR   : GOTO TOP OF TEST?
1646 007050 001004                BNE      1S          : BR IF YES
1647 007052 032777 002000 172160     BIT      #SM10,2SMR  : GOTO NEXT TEST?
1648 007060 001400                BEQ      2S          : BR IF NO
1649                    .MOV      NEXT,$LPADR  : SET FOR NEXT TEST
1650 007062                .MOV      $STACK,SP  : RESET SP
1651                    JMP      2$LPADR    : GOTO SPECIFIED TEST
1652                    RTI                    : $LPADR
1653                    ERTAB0: 1
1654                    .BYTE      6,2
1655                    ERTAB1: $AVPC
1656                    XTSTN: 1
1657                    .BYTE      3,2
1658                    $STSTN
1659                    ;ENTER HERE ON POWER FAILURE
1660
1661
1662
1663
1664
1665
1666                    .SBTTL  POWER DOWN AND UP ROUTINES
1667
1668                    ;*****
1669                    ;POWER DOWN ROUTINE
1670                    $PWDRN: MOV      $SILLUP,$#PWVREC  : SET FOR FAST UP
1671                    MOV      $340,$#PWVREC+2 : $RIO:7
1672                    MOV      RO,-(SP)      : PUSH RO ON STACK
1673                    MOV      R1,-(SP)      : PUSH R1 ON STACK
1674                    MOV      R2,-(SP)      : PUSH R2 ON STACK
1675                    MOV      R3,-(SP)      : PUSH R3 ON STACK
1676                    MOV      R4,-(SP)      : PUSH R4 ON STACK
1677                    MOV      R5,-(SP)      : PUSH R5 ON STACK
1678                    MOV      2SMR,-(SP)    : PUSH 2SMR ON STACK
1679                    MOV      SP,$$AVR6     : SAVE SP
1680                    MOV      $#PWUP,$#PWVREC ;; SET UP VECTOR
1681                    HALT
1682                    BR          .-2          ;; HANG UP
1683
1684                    ;*****

```

POWER DOWN AND UP ROUTINES

```

1681      :POWER UP ROUTINE
1682      SPMRUP:  MOV    #SILLUP,2#PMRVEC  ;SET FOR FAST DOWN
1683      MOV    $SAVR6,SP  ;GET SP
1684      CLR    $SAVR6      ;WAIT LOOP FOR THE TTY
1685      IS:   INC    $SAVR6  ;WAIT FOR THE INC
1686      BNE   IS          ;OF WORD
1687      TYPE  ,MPFAIL
1688      CNVRT ,PFTAB
1689      CLAB  $SERFLG      ;CLEAR ERROR FLAG.
1690      CLR  $SERAPC      ;CLEAR LAST ERROR PC
1691      MOV  $KMC SR,R1   ;RESTORE DEVICE ADDRESS.
1692      CLR  (R1)         ;CLEAR THE CSR.
1693      MSTCLR
1694      MOV  (SP)+,2$SR   ;POP STACK INTO 2$SR
1695      MOV  (SP)+,R5     ;POP STACK INTO R5
1696      MOV  (SP)+,R4     ;POP STACK INTO R4
1697      MOV  (SP)+,R3     ;POP STACK INTO R3
1698      MOV  (SP)+,R2     ;POP STACK INTO R2
1699      MOV  (SP)+,R1     ;POP STACK INTO R1
1700      MOV  (SP)+,R0     ;POP STACK INTO R0
1701      MOV  #SPMRDN,2#PMRVEC ;SET UP THE POWER DOWN VECTOR
1702      MOV  #340,2#PMRVEC+2 ;PRIO:7
1703      TYPE  ,WORD      ;REPORT THE POWER FAILURE
1704      SPWRMG: .WORD  MPFAIL ;POWER FAIL MESSAGE POINTER
1705      RTI
1706      SILLUP: HALT
1707      BR    .-2        ;THE POWER UP SEQUENCE WAS STARTED
1708      $SAVR6: 0         ;BEFORE THE POWER DOWN WAS COMPLETE
1709                        ;PUT THE SP HERE
1710      PFTAB: 1
1711      .BYTE  3,2
1712      $STSTNM
1713
1714      .DELAY:
1715      MOV    #20,2$KMP04
1716      ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
1717      121111 ;POKE CLOCK DELAY BIT
1718      IS:
1719      ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
1720      121224 ;PORT4+IBUS#11
1721      BIT    #BIT4,2$KMP04 ;IS CLOCK BIT SET?
1722      BEQ   IS         ;BR IF NO
1723      RTI
1724
1725      .MSTCLR:
1726      BISB  #BIT6,2$KMC SRH ;SET MASTER CLEAR
1727      BICB  #BIT6!BIT7,2$KMC SRH ;CLEAR MASTER CLEAR AND RUN
1728      RTI ;RETURN
1729
1730      .ROMCLK:
1731      BISB  #BIT1,2$KMC SRH ;SET ROMI
1732      MOV  2(SP)+,2$KMP06 ;LOAD INSTRUCTION IN SEL6
1733      ADD  #2,-(SP) ;ADJUST STACK
1734      BIT  #SM06,2$SR ;HALT IF SM06 =1
1735      BEQ  IS ;BR IF SM06 =0
1736      HALT ;HALT BEFORE CLOCKING INSTRUCTION

```

POWER DOWN AND UP ROUTINES

```

1737 007402 152777 000003 172460 1S: BISC #BIT1!BIT0,2KMC5RH ;CLOCK INSTRUCTION
1738 007410 142777 000007 172452 BICB #BIT2!BIT1!BIT0,2KMC5RH ;CLEAR ROAD, ROMI, STEP
1739 007416 000002 RTI

1740
1741 007420 .DATACLK:
1742 007420 013637 011112 MOV #2(SP)+,TEMP ;PUT TICK COUNT IN TEMP
1743 007424 062746 000002 ADD #2,-(SP) ;ADJUST STACK
1744 007430 152777 000020 172432 1S: BISC #BIT4,2KMC5RH ;SET STEP LU
1745 007436 027777 172424 172422 CMP #2KMC5R,2KMC5R ;WASTE TIME
1746 007444 142777 000020 172416 BICB #BIT4,2KMC5RH ;CLEAR STEP LU
1747 007452 005337 011112 DEC TEMP ;DEC TICK COUNT
1748 007456 001364 BNE IS ;BR IF NOT DONE
1749 007460 000002 RTI ;RETURN
1750 007462 000001 3S: .BLKW 1

1751
1752 007464 .TIMER:
1753 007464 013637 011112 MOV #2(SP)+,TEMP ;MOVE COUNT TO TEMP
1754 007470 062746 000002 ADD #2,-(SP) ;ADJUST STACK
1755 007474 1S:
1756 007474 104412 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
1757 007476 021364 021364 ;PORT4+IBUS# REG11
1758 007500 032777 000002 172366 BIT #2,2KMP04 ;IS PGM CLOCK BIT CLEAR?
1759 007506 001772 BEO IS ;BR IF YES
1760 007510 2S:
1761 007510 104412 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
1762 007512 021364 021364 ;PORT4+IBUS# REG11
1763 007514 032777 000002 172352 BIT #2,2KMP04 ;IS PGM CLOCK BIT SET?
1764 007522 001372 BNE 2S ;BR IF YES
1765 007524 005337 011112 DEC TEMP ;DEC COUNT
1766 007530 001364 BNE IS ;BR IF NOT DONE
1767 007532 000002 RTI ;RETURN
1768
1769 007534 050200 051127 043040 MPFAIL: .ASCIZ <200>/PWR FAILED. RESTART AT TEST /
(2) 007572 042600 042116 050040 MEPASS: .ASCIZ <200>/END PASS DZKCA /
(2) 007614 051200 000 MR: .ASCIZ <200>/R/
(2) 007617 200 047516 042040 MERR2: .ASCIZ <200>/NO DEVICES PRESENT./
(2) 007644 044600 051516 043125 MERR3: .ASCIZ <200>/INSUFFICIENT DATA!./
(2) 007670 046200 041517 020113 MLOCK: .ASCIZ <200>/LOCK ON SELECTED TEST/
(2) 007717 103 051123 020072 MCSR: .ASCIZ /CSR: /
(2) 007725 126 041505 020072 MVECK: .ASCIZ /VEC: /
(2) 007733 128 051501 042523 MPASSX: .ASCIZ /PASSES: /
(2) 007744 051105 047522 051522 MERRX: .ASCIZ /ERRORS: /
(2) 007755 124 051505 020124 NTSTN: .ASCIZ /TEST NO: /
(2) 007767 052 MASTEK: .ASCIZ /#/
(2) 007771 200 042523 020124 MNEM: .ASCIZ <200>/SET SWITCH REG TO KMC11'S DESIRED ACTIVE./
(2) 010044 041520 020072 000 MERRPC: .ASCIZ /PC: /
(2) 010051 200 020040 020040 XHEAD: .ASCIZ <200>/
(2) 010110 020200 020040 020040 .ASCIZ <200>/
(2) 010147 200 020040 041520 .ASCIZ <200>/ PC CSR STAT1 STAT2 STAT3/
(2) 010221 200 026455 026455 .ASCIZ <200>/-----
(2) 010275 200 047510 020127 NUM: .ASCIZ <200>/HOW MANY KMC11'S TO BE TESTED?/
(2) 010335 200 051503 0 CSR: .ASCIZ <200>/CSR ADDRESS?/
(2) 010353 200 042526 0 VEC: .ASCIZ <200>/VECTOR ADDRESS?/
(2) 010374 041200 020122 0 PRIO: .ASCIZ <200>/BR PRIORITY LEVEL? (4,5,6,7)?/
(2) 010433 200 044127 0 .ASCIZ <200>/WHICH LINE UNIT? IF NONE TYPE "N", IF M8201 TYPE "1", IF M8202 TYP
(2) 010545 200 053520 0 .ASCIZ <200>/SWITCH PAC#1 (DDCMP LINE #)?/

```

POWER DOWN AND UP ROUTINES

```

(2) 010603 200 052523 052111 BR: .ASCIZ <200>/SWITCH P&C#2 (BM873 BOOT #00)?/
(2) 010643 200 051511 052040 CONN: .ASCIZ <200>/IS THE LOOP BACK CONNECTOR ON?/
(2) 010703 200 047516 042040 NOACT: .ASCIZ <200>/NO DEVICES ARE SELECTED/
(2) 010734 100200 046513 030503 CONERR: .ASCIZ <200><200>/KMC11 AT NONSTANDARD ADDRESS PC: /
(2) 011001 200 054105 042520 CENR: .ASCIZ <200>/EXPECTED FOUND/
(2) 011022 024040 046513 024503 KMC1: .ASCIZ / (KMC) /
(2) 011032 044600 046114 043505 MINT: .ASCIZ <200>/ILLEGAL INTERRUPT ERROR/
(2) 011064 000005 .EVEN
(2) 011064 000005 XSTAT0: 5
1770 011066 006 003 .BYTE 6,3
1771 011070 001276 $TMP0
1772 011072 006 003 .BYTE 6,3
1773 011074 001300 $TMP1
1774 011076 006 003 .BYTE 6,3
1775 011100 001302 $TMP2
1776 011102 006 003 .BYTE 6,3
1777 011104 001304 $TMP3
1778 011106 006 002 .BYTE 6,2
1779 011110 001306 $TMP4
1780 .EVEN
1781 ;BUFFERS FOR INPUT-OUTPUT
1782
1783
1784 011112 000000 TEMP: 0
1785 011154 .=. +40
1786 011154 000000 MDATA: 0
1787 011216 .=. +40
1788
1789
1790 ;ROUTINE USED TO CHANGE SOFTWARE SWITCH
1791 ;REGISTER USING THE CONSOLE TERMINAL
1792
1793 -----
1794 011216 022737 000176 001240 CKSMR: CMP #SMREG, SMR ; IS THE SOFT SMR BEING USED?
1795 011224 001075 BNE CKSMRS ; BR IF NO
1796 011226 132737 000001 001336 BITB #1, SMV ; IS IT RUNNING UNDER APT?
1797 011234 001071 BNE CKSMRS ; EXIT IF YES.
1798 011236 022777 000007 170002 CMP #7, 2STKB ; WAS CTRL G TYPED? (7 BIT ASCII)
1799 011244 001404 BEQ #1 ; BR IF YES
1800 011246 022777 000207 167772 CMP #207, 2STKB ; WAS CTRL G TYPED? (8 BIT ASCII)
1801 011254 001061 BNE CKSMRS ; BR IF NO
1802 011256 010246 IS: MOV R2, -(SP) ; STORE R2
1803 011260 010346 MOV R3, -(SP) ; STORE R3
1804 011262 010446 MOV R4, -(SP) ; STORE R4
1805 011264 012737 177777 011422 MOV #1, SMFLG ; SET SOFT TYPE OUT FLAG
1806 011272 005002 CKSMR1: CLR R2 ; CLEAR NEW SMR CONTENTS
1807 011274 012704 177777 MOV #1, R4 ; SET FLAG TO ALL ONES
1808 011300 104401 005531 TYPE , SMR ; TYPE "SMR: "
1809 011304 104417 CKSMR2: CNVRT ; TYPE OUT PRESENT CONTENTS
1810 011306 011456 SOFTSW OF SOFT SWITCH REGISTER
1811 011310 104401 005542 CKSMR3: TYPE #MNEW ; TYPE "NEW? "
1812 011314 004737 011424 CKSMR4: JSR PC, INCHAR ; GET RESPONSE
1813 011320 022703 000015 CMP #15, R3 ; WAS IT A CR?
1814 011324 001424 BEQ $S ; BR IF YES
1815 011326 022703 000012 CMP #12, R3 ; WAS IT A LF?
1816 011332 001416 BEQ $S ; BR IF YES

```


1817	011334	022703	000025			CHP	025,R3	: WAS IT CTRL U?
1818	011340	001754				BEG	CKSMR1	: BR IF YES(START OVER)
1819	011342	022703	000007			CHP	07,R3	: IF CNTL G GET NEXT CHAR
1820	011344	001754				BEG	CKSMR4	
1821	011350	005004				CLR	R2	: IT MUST BE A DIGIT SO CLR FLAG
1822	011352	042703	177770			BIC	0177770,R3	: ONLY 0-7 ARE LEGAL SO MASK OFF BITS
1823	011354	005302				R2		: SHIFT R2 3 TIMES
1824	011356	005302				R2		
1825	011358	005302				R2		
1826	011360	005302				R2		
1827	011362	001754				BIS	CKSMR	: ADD LAST DIGIT
1828	011364	001754				BR	CKSMR	: GET NEXT CHARACTER
1829	011370	012766	002402	000006	4S:	MOV	0,START,6(SP)	: LF WAS TYPED SO GO TO START
1830	011376	005704			5S:	TST	R2	: IS FLAG CLEAR?
1831	011400	001002				BR	R2	: IF NOT DON'T CHANGE SOFT SWR
1832	011406	010277	167632			MOV	0,SWR	: IF YES THEN WRITE NEW CONTENTS TO SOFT SWR
1833	011412	005037	011422		6S:	CLR	SMFLG	: CLEAR TYPEOUT FLAG
1834	011418	012604				MOV	(SP)+,R4	: RESTORE R4
1835	011414	012603				MOV	(SP)+,R3	: RESTORE R3
1836	011416	012602				MOV	(SP)+,R2	: RESTORE R2
1837	011420	000207			CKSMRS:	RTS	PC	: RETURN
1838	011422	000000			SMFLG:	0		
1839								
1840	011424	105777	167614		INCHAR:	TSTB	0STKS	
1841	011430	100375				BPL	-4	
1842	011432	017703	167610			MOV	0STKB,R3	
1843	011436	105777	167606			TSTB	0STPS	
1844	011442	100375				BPL	-4	
1845	011444	010377	167602			MOV	R3,0STPB	
1846	011450	042703	000200			BIC	0017,R3	
1847	011454	000207				RTS	PC	
1848								
1849	011456	000001			SOFTSW:	1		
1850	011460	006	002			BYTE	6,2	
1851	011462	000176				SMFLG		

POWER DOWN AND UP ROUTINES

1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907

011464	005737	001470	
011470	001004		
011472	104401	010703	
011476	000000		
011500	000776		
011502	000241		
011504	005137	001500	
011510	005137	001500	
011514	002737	000004	001504
011522	002737	000010	001502
011530	002737	002300	001502
011536	001006		
011540	012737	002100	001502
011546	012737	002302	001504
011554	003737	001500	001470
011562	001747		
011564	013700	001502	
011570	013702	001504	
011574	012037	002066	
011600	011037	002056	
011604	042737	177000	002056
011612	012037	002050	
011616	012037	002052	
011622	012037	002054	
011628	012237	001324	
011632	012237	001212	
011636	012700	000002	
011642	013737	002066	002070
011650	005237	002070	
011654	013737	002070	002072
011662	005237	002072	
011666	013737	002072	002074
011674	060037	002074	
011700	013737	002074	002076
011706	060037	002076	
011712	013737	002056	002060
011720	060037	002060	
011724	013737	002060	002062
011732	060037	002062	
011736	013737	002062	002064
011744	060037	002064	
011750	012777	034664	170100
011756	012777	000240	170074
011764	012777	035360	170070
011772	012777	000240	170064
012000	012737	000200	177776

```

ROUTINE USED TO "CYCLE" THROUGH UP TO 16 KMC11'S
THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
AND RUNS THE SPECIFIED KMC11'S. THIS ROUTINE #MUST#
BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
SETUP NECESSARY.

```

CYCLE:	TST	KNACTV		ARE ANY KMC11'S TO BE TESTED?
	BNE	IS		BR IF OK.
	TYPE	,NOACT		NO KMC11'S SELECTED!!
	HALT			STOP THE SHOW.
	BR	.-2		DISQUALIFY CONT. SW.
15:	CLC			CLEAR PROC. CARRY BIT.
	ROL	RUN		UPDATE POINTER
	ADC	RUN		CATCH CARRY FROM RUN
	ADD	#4,MILK		UPDATE POINTER
	ADD	#10,CREAM		UPDATE ADDRESS POINTER.
	CMP	#KH.MAP+200,CREAM		
	BNE	25		KEEP GOING; NOT ALL TESTED FOR.
	MOV	#KH.MAP,CREAM		RESET ADDRESS POINTER.
	MOV	#CNT.MAP,MILK		RESET PASS COUNT POINTER
25:	BIT	RUN,KNACTV		IS THIS ONE ACTIVE?
	BEQ	IS		BR IF NO
	MOV	CREAM,R0		GET ADDRESS POINTER
	MOV	MILK,R2		GET PASS COUNT POINTER
	MOV	(R0)+,KNCSR		LOAD SYSTEM CTRL. REG
	MOV	(R0),KNRVEC		LOAD VECTOR
	BIC	#177000,KNRVEC		CLEAR UNWANTED BITS
	MOV	(R0)+,STAT1		LOAD STAT1
	MOV	(R0)+,STAT2		LOAD STAT2
	MOV	(R0)+,STAT3		LOAD STAT3
	MOV	(R2)+,SPASS		LOAD PASS COUNT
	MOV	(R2)+,SERCTL		LOAD ERROR COUNT
	MOV	#2,R0		SAVE CORE THIS WAY!
	MOV	KNCSR,KNCSRH		
	INC	KNCSRH		
	MOV	KNCSRH,KNCTL		
	INC	KNCTL		
	MOV	KNCTL,KMP04		
	ADD	R0,KMP04		
	MOV	KMP04,KMP06		
	ADD	R0,KMP06		
	MOV	KNRVEC,KMRLVL		PTY LVL
	ADD	R0,KMRLVL		
	MOV	KMRLVL,KMTVEC		TX VEC
	ADD	R0,KMTVEC		
	MOV	KMTVEC,KMTLVL		TX LVL
	ADD	R0,KMTLVL		
	MOV	#INT,INTVEC		SET UP INTERRUPT VECTORS.
	MOV	#PRM,PRMVEC		SET PRIORITY 5.
	MOV	#OUI,OUIVEC		SET UP INTERRUPT VECTORS.
	MOV			SET PRIORITY 5.
	MOV			SET MAIN PROGRAM PRIORITY AS 4.

POWER DOWN AND UP ROUTINES

```

1908 ;SOTHAT KMC CAN INTERRUPT...
1909
1910 012006 032737 000002 001446 BIT #SW01,STRTSW ;IS TEST NO. SELECTED
1911 012014 001447 BEQ 75 ;BR IF NO
1912 4S: 012016 TST 2842 ;RUNNING IN AUTO MODE?
1913 012016 005737 000042 BNE 75 ;BR IF YES
1914 012022 001044 TYPE ,SCRLF
1915 012024 104401 001313 INPUT
1916 012030 104415 NTSTN
1917 012032 007755 1
1918 012034 000001 1000
1919 012036 001000 $STSTM
1920 012038 001202 0
1921 012040 000 .BYTE
1922 012042 001 .BYTE
1923 012044 012700 013774 MOV #TST1,RO
1924 012050 022710 5S: CMP (PC)+,(RO) ;CMP FIRST WORD TO 12737
1925 012052 012737 MOV (PC)+,2(PC)+
1926 012054 001020 BNE 6S ;BR IF NOT SAME
1927 012056 023760 001202 000002 CMP $STSTM,2(RO) ;DOES $STSTM MATCH?
1928 012058 001014 BNE 6S ;BR IF NO
1929 012060 022760 001202 000004 CMP #STSTM,4(RO) ;IS LAST WORD OK?
1930 012074 001010 BNE 6S ;BR IF NO
1931 012076 010037 001206 MOV RO,SLPADR ;IT IS A LEGAL TEST SO DO IT
1932 012102 104401 007614 TYPE #R
1933 012106 042737 000002 001446 BIC #SW01,STRTSW
1934 012114 000412 BR 8S
1935 012116 005730 6S: TST (RO)+ ;POP RO
1936 012120 020027 034324 CMP RO,#TLAST+10 ;AT END YET?
1937 012124 001351 BNE 5S ;BR IF NO
1938 012126 104401 001312 TYPE #QUES ;YES ILLEGAL TEST NO.
1939 012132 000731 BR 4S ;TRY AGAIN
1940
1941 012134 012737 013774 001206 7S: MOV #TST1,SLPADR ;PREPARE SLPADR ADDRESS
1942 012142 013701 002066 8S: MOV KMC11,R1 ;R1 = BASE KMC11 ADDRESS
1943 012146 000177 167034 JMP 2SLPADR ;GO START TESTING.
1944
1945
1946
1947 ;ROUTINE USED TO "AUTO SIZE" THE KMC11
1948 ;CSR AND VECTOR.
1949 ;NOTE: THE CSR MAY BE ANY WHERE IN THE FLOATING
1950 ;ADDRESS RANGE (16000:16400)
1951 ;AND THE VECTOR MAY BE ANY WHERE IN THE
1952 ;FLOATING VECTOR RANGE (300:770)
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
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

```

```

AUTO.SIZE:
CSRMAP: MOV #KH.MAP,R2 ;INSURE A BUS INIT.
1S: CLR (R2)+ ;LOAD MAP POINTER.
CMP #KH.END,R2 ;ZERO ENTIRE MAP
BNE 1S ;ALL DONE?
CLR KMINUM ;BR IF NO
MOV #KH.MAP,R2 ;SET OCTAL NUMBER OF KMC11'S TO 0
CLR KMACTV ;R2 POINTS TO KMC MAP
BIT #SW00,STRTSW ;CLEAR ACTIVE
;QUESTIONS?

```

P05

```

1964 012212 001002 BNE #6 ;BR IF YES
1965 012214 000137 JMP #7 ;IF NO SKIP QUESTIONS
1966 012216 012737 000001 001306 MOV #1,STMP4 ;START WITH 1
1967 012218 104415 INPUT NUM
1968 012220 010275 1
1969 012222 000001 16
1970 012224 000020 STMP2
1971 012226 001302 .BYTE 0
1972 012228 000 .BYTE 1
1973 012230 001 .MOV STMP2,KMNUM ;KMNUM = HOW MANY
1974 012232 013737 001302 001472 ,SCRLF ;TYPE WHICH KMC IS BEING DONE
1975 012234 104401 001313 12S: CONVRT ;STMP4 IS WHICH KMC
1976 012236 104415 WHICH
1977 012238 013223 INC STMP4
1978 012240 006237 001306 INPUT CSR
1979 012242 104415 16000
1980 012244 010275 164000
1981 012246 164000 STMP3
1982 012248 001304 .BYTE 0
1983 012250 000 .BYTE 1
1984 012252 001 .MOV STMP3,(R2)+ ;STORE CSR IN MAP
1985 012254 013722 001304 INPUT VEC
1986 012256 104415 0
1987 012258 010353 776
1988 012260 000000 STMP3
1989 012262 000776 .BYTE 0
1990 012264 001304 .BYTE 1
1991 012266 000 STMP3,(R2) ;STORE VECTOR IN MAP
1992 012268 013712 001304 NOV
1993 012270 104401 10S: TYPE
1994 012272 010374 PRIO ;ASK WHAT BR LEVEL
1995 012274 004737 013520 JSR PC,INTTY ;GET RESPONSE
1996 012276 022703 000024 CMP #24,R3
1997 012278 101014 BHI #505 ;BR IF LESS THAN 4
1998 012280 022703 000027 CMP #27,R3
1999 012282 103411 BLO #505 ;BR IF GREATER THAN 7
2000 012284 012704 000011 MOV #11,R4 ;R4 = NUMBER OF SHIFTS
2001 012286 006303 RSL ;SHIFT R3 LEFT
2002 012288 005304 DEC R4 ;DEC SHIFT COUNT
2003 012290 001375 BNE #-4 ;BR IF NOT DONE
2004 012292 042703 170777 BIC #170777,R3 ;BIC UNWANTED BITS
2005 012294 050312 BIS R3,(R2) ;PUT BR LEVEL IN STATUS MAP
2006 012296 000403 BR #5 ;CONTINUE
2007 012298 104401 50S: TYPE
2008 012300 001312 SQUES ;RESPONSE IS OUT OF LIMITS
2009 012302 000752 BR #10S ;TRY AGAIN
2010 012304 104401 8S:
2011 012306 010433 9S:
2012 012308 004737 013520 16S: TYPE
2013 012310 022703 000021 JSR PC,INTTY ;ASK WHICH LINE UNIT
2014 012312 001417 BEQ #30S ;GET REPLY
2015 012314 022703 000022 CMP #22,R3 ;"1"
2016 012316 022703 000022 CMP #22,R3 ;"2"

```

POWER DOWN AND UP ROUTINES

2020	012422	001412			BEG	315		
2021	012424	022703	000116		CHP	0116,R3		; "N"
2022	012430	001403			BEG	325		
2023	012430	104401			TYPE			
2024	012439	001312			SOLES			; IF NOT A 1,2 OR N TYPE "?"
2025	012436	000760			BR	165		; TRY AGAIN
2026	012439	052722	010000	325:	BIS	#BIT12,(R2)+		; SET BIT 12 IN STAT2 IF NO LU
2027	012441	022722			CHP	(R2)+,(R2)+		; POP OVER STAT2 AND STAT3
2028	012446	000745			BR	335		
2029	012460	052712	020000	315:	BIS	#BIT13,(R2)		; SET BIT 13 IN STAT2 IF M8202
2030	012464	104401		305:	TYPE			
2031	012466	010643			CONN			; ASK IF LOOP-BACK IS ON
2032	012460	004737	013520		JSR	PC,INTTY		; GET REPLY
2033	012464	022703	000131		CHP	0131,R3		; Y
2034	012470	001406			BEG	175		
2035	012472	022703	000116		CHP	0116,R3		; N
2036	012476	001406			BEG	185		
2037	012500	104401			TYPE			
2038	012502	001312			SOLES			; IF NOT Y OR N TYPE "?"
2039	012504	000763			BR	305		; TRY AGAIN
2040	012506	052722	040000	175:	BIS	#BIT14,(R2)+		; TURNAROUND IS CONNECTED
2041	012512	000402			BR	195		
2042	012514	042722	040000	185:	BIC	#BIT14,(R2)+		; NO TURNAROUND
2043	012520			195:				
2044	012520	104415			INPUT			
2045	012522	010545			LINE			
2046	012524	000000			0			
2047	012526	000377			377			
2048	012530	001304			STMP3			
2049	012532	000			.BYTE	0		
2050	012533	001			.BYTE	1		
2051	012534	113722	001304		MOVB	STMP3,(R2)+		; STORE SWITCH PAC IN MAP
2052	012534	104415			INPUT			
2053	012534	010603			BM			
2054	012534	000000			0			
2055	012534	000377			377			
2056	012534	001304			STMP3			
2057	012534	000			.BYTE	0		
2058	012534	001			.BYTE	1		
2059	012534	113722	001304		MOVB	STMP3,(R2)+		; STORE SWITCH PAC IN MAP
2060	012550	005722			TST	(R2)+		; POP OVER STAT3
2061	012552	005337	001302	335:	DEC	STMP2		; DEC KMC COUNT
2062	012556	001230			BNE	125		; BR IF MORE TO DO
2063	012570	000137	013126		JMP	135		; CONTINUE
2064	012574	012701	160000	75:	MOV	#160000,R1		; SET FOR FIRST ADDRESS TO BE TESTED
2065	012580	012737	013220	000004	MOV	#65,2#4		; SET FOR NON-EXISTANT DEVICE TIME OUT
2066	012586	005011		25:	CLR	(R1)		; CLEAR SEL0
2067	012510	005711			TST	(R1)		; IF KMC11 KMC SR S/B 0
2068	012612	001135			BNE	35		; IF NO DEV ; TRAP TO 4. IF NO BIT 8 THEN NO KMC11
2069	012614	005061	000006		CLR	6(R1)		; CLEAR SEL6
2070	012620	005761	000006		TST	6(R1)		; IF KMC11 THEN KMRIC S/B =0!
2071	012624	001130			BNE	35		; BR IF NOT KMC11
2072	012626	012711	002000		MOV	#BIT10,(R1)		; SET ROM0
2073	012632	005061	000004		CLR	4(R1)		; CLEAR SEL4
2074	012636	012761	125252	000006	MOV	#125252,6(R1)		; WRITE THIS TO SEL6
2075	012644	052711	020000		BIS	#BIT13,(R1)		; WRITE IT!

```

012660 002761 125252 000004      CMP      #125252,4(R1)      ; WAS IT WRITTEN?
012666 001113                      BNE      35                ; IF NO IT IS NOT CRAM
; AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A KMC11 CSR ADDRESS.
012669 010122                      ; STORE CSR IN CORE TABLE.
012671 001000      MOV      R1,(R2)+          ; CLEAR LINE UNIT LOOP
012673 000004      MOV      #BIT9,(R1)       ; CLEAR PORT4
012675 122113 000006      CLR      4(R1)           ; CLEAR PORT4
012677 000400      MOV      #122113,6(R1)   ; LOAD INSTRUCTION (CLR DTR)
012679 001264 000006      BIS      #BIT8,(R1)     ; CLOCK INSTRUCTION
012681 000400      MOV      #021264,6(R1)  ; LOAD INSTRUCTION
012683 000377 000004      BIS      #BIT8,(R1)     ; CLOCK INSTRUCTION
012685 001000      CMPB   #377,4(R1)       ; IS IT ALL ONES?
012687 010000      BNE      .+10           ; BR IF NO
012689 000002 000004      BIS      #BIT12,(R2)    ; IF YES, NO LINE UNIT, SET STATUS BIT
012691 000002 000004      BR      205            ; IS SWITCH A ONE?
012693 060000      BIT     #BIT1,4(R1)     ; BR IF #B01
012695 000010 000004      BIS     #BIT13:BIT14,(R2) ; #B002 ASSUME CONNECTOR
012697 000010 000004      BR      205            ; CONNECTOR ON)
012699 000010 000004      BIT     #BIT3,4(R1)     ; IS #B01 SET
012701 000100 000004      BNE     205            ; BR IF #B01 NO CONNECTOR (ON LINE)
012703 122113 000006      MOV     #122113,6(R1)   ; LOAD INSTRUCTION
012705 000400      BIS     #BIT8,(R1)     ; CLOCK INSTRUCTION (SET DTR)
012707 001264 000006      MOV     #021264,6(R1)  ; LOAD INSTRUCTION
012709 000400      BIS     #BIT8,(R1)     ; CLOCK INSTRUCTION (READ MODEM REG)
012711 000010 000004      BIT     #BIT3,4(R1)     ; IS #B01 SET NOW?
012713 040000      BEQ     205            ; BR IF NO CONNECTOR
012715 000000      BIS     #BIT14,(R2)    ; SET STATUS BIT FOR CONNECTOR
012717 021324 000006 205:   TST     (R2)+           ; POP POINTER
012719 001400      MOV     #021324,6(R1)   ; PUT INSTRUCTION IN PORT6
012721 000004      MOV     #BIT9:BIT8,(R1) ; PORT4+LU IS
012723 001344 000006      BISB   4(R1),(R2)+     ; STORE DOCHP LINE # IN TABLE
012725 001400      MOV     #021344,6(R1)  ; PORT6+INSTRUCTION
012727 000004      BISB   4(R1),(R2)+     ; CLOCK INSTR.
012729 000004      TST     (R2)+           ; STORE #B073 ADD IN TABLE
012731 001472      CLR     (R1)           ; POP OVER STAT3
012733 000020 001472      INC     KNUM           ; CLEAR #011
012735 001410      CMP     #20,KNUM       ; UPDATE DEVICE COUNTER
012737 000011      BEQ    135            ; ARE MAX. NO. OF DEV FOUND?
012739 000006      CLR     (R1)         ; YES DON'T LOOK FOR ANY MORE.
012741 000010      CLR     6(R1)        ; CLEAR BIT 10
012743 164000      ADD     #10,R1        ; CLEAR SEL 6
012745 001470      CMP     #164000,R1    ; UPDATE CSR POINTER ADDRESS
012747 001472      BNE     25            ; BR IF MORE ADDRESS TO CHECK.
012749 001470      CLR     KMACTV       ; HERE ANY KMC11'S FOUND AT ALL?
012751 001472      TST     KNUM         ; ERROR AUTO SIZER FOUND NO KMC11'S IN THIS SYS.
012753 001423      BEQ    55            ; SAVE NUMBER OF DEVICES
012755 001472      MOV     KNUM,R1      ; GENERATE ACTIVE REGISTER OF DEVICES.
012757 001476      MOV     R1,SAVNUM    ; SET THE BIT
012759 000241 45:        CLC                    ; BR IF MORE TO GENERATE
012761 001470      ROL     KMACTV
012763 005237      INC     KMACTV
012765 005301      DEC     R1
012767 001371      BNE     45

```


POWER DOWN AND UP ROUTINES

2132	013166	012737	000006	000004		MOV	#6, #4	:RESTORE TRAP VECTOR
2133	013174	013737	001470	001474		MOV	KMACTV, SAVACT	:SAVE ACTIVE REGISTER
2134	013202	000137	013234			JMP	VECMAP	:GO FIND THE VECTOR NOW.
2135	013206	104401	007617		55:	TYPE	MERR2	:NOTIFY OPR THAT NO KMC11'S FOUND.
2136	013212	005000				CLR	R0	:MAKE DATA LIGHTS ZERO
2137	013214	000000				HALT		:STOP THE SHOW
2138	013216	000776				BR	.-2	:DISABLE CONT. SW.
2139	013220	012716	013114		65:	MOV	#145, (SP)	:ENTERED BY NON-EXISTANT TIME-OUT.
2140	013224	000002				RTI		:RETURN TO MAINSTREAM
2141								
2142	013226	000001			44TC4.	1		
2143	013230	002	0			.BYTE	2,2	
2144	013232	001306				STMP4		
2145								
2146	013234	032737	000001	001	ECMAP:	BIT	#SM00, STRTSM	
2147	013242	001114				BNE	55	
2148	013244	012737	000340	000		MOV	#340, #22	:SET IOT TRAP PRIO TO 7
2149	013250	012737	013426	00002C		MOV	#45, #20	:SET IOT TRAP VECTOR
2150	013252	012702	002100			MOV	#CM, MAP, R2	:SET SOFTWARE POINTER
2151	013254	012700	000300			MOV	#300, R0	:FLOATING VECTORS START HERE.
2152	013270	012701	000302			MOV	#302, R1	:PC OF IOT INSTR.
2153	013274	010120			15:	MOV	R1, (R0)+	:START FILLING VECTOR AREA
2154	013276	012721	000004			MOV	#4, (R1)+	:WITH +2; IOT
2155	013302	022021				CMP	(R0)+, (R1)+	:ADD 2 TO R0 +R1
2156	013304	020127	001000			CMP	R1, #1000	
2157	013310	101771				BLOS	15	:BR IF MORE TO FILL
2158	013312	013737	001470	001276		MOV	KMACTV, STMP0	:STORE TEMPORALLY
2159	013320	005037	001276		25:	ROR	STMP0	:BRING OUT A BIT
2160	013324	103063				BCC	55	:BR IF ALL DONE
2161	013326	012704	000012			MOV	#12, R4	:R4 IS INDEX REGISTER
2162	013328	016437	013504	177776		MOV	BR, VL(R4), PS	:SET PS TO 7
2163	013330	011201				MOV	(R2), R1	
2164	013332	012761	000200	000004		MOV	#200, 4(R1)	
2165	013334	012711	001000			MOV	#BIT9, (R1)	:SET ROMI
2166	013336	012761	121111	000006		MOV	#121111, 6(R1)	:PUT INSTRUCTION IN PORT6
2167	013338	012711	001400			MOV	#BIT9:BIT8, (R1)	:FORCE AN INTERRUPT
2168	013342	105200			75:	INCB	R0	:STALL
2169	013370	001376				BNE	.-2	:FOR TIME TO INTERUPT
2170	013372	162704	000002			SUB	#2, R4	:GET NEXT LOWEST PS LEVEL
2171	013376	001404			65:	BEG	65	:BR IF R4 = 0
2172	013400	016437	013504	177776		MOV	BR, VL(R4), PS	:MOVE NEXT LOWER LEVEL IN PS
2173	013402	000767				BR	75	:BR TO DELAY
2174	013410	022762	005300	000002	65:	BIS	#5300, 2(R2)	:NO INTERRUPT ASSUME 300 AT LEVEL 5 AND FIX KMC11 LATER
2175	013416	005011			35:	CLR	(R1)	:CLEAR ROMI
2176	013420	022702	000010			ROD	#10, R2	:POP SOFTWARE POINTER
2177	013424	000736				BR	25	:KEEP GOING
2178	013426	051662	000002		45:	BIS	(SP), 2(R2)	:GET VECTOR ADDRESS
2179	013428	042762	000007	000002		BIC	#7, 2(R2)	:CLEAR JUNK
2180	013430	015405				MOV	BR, VL+2(R4), R5	:GET BR LEVEL OF KMC11
2181	013434	005305				RSL	R5	:SHIFT LEVEL 4 PLACES
2182	013436	005305				RSL	R5	:TO THE LEFT FOR THE
2183	013438	005305				RSL	R5	:STATUS TABLE
2184	013440	005305				RSL	R5	
2185	013454	042705	170777			BIC	#170777, R5	:CLEAR UNWANTED BITS
2186	013460	050562	000002			BIS	R5, 2(R2)	:PUT BR LEVEL IN STATUS TABLE
2187	013464	022626				CMP	(SP)+, (SP)+	:POP IOT JUNK OFF STACK

218	013746	012716	013416	MOV	#35, (SP)	;SET FOR RETURN
219	013747	000002		RTI		
220	013748	012737	004134 000020 55:	MOV	#SCOPE, #20	; RESTORE SCOPE VECTOR
221	013749	000207		RTS	PC	; ALL DONE WITH "AUTO SIZING"
222	013750	000000		BRLVL:	PR0	; LEVEL 0
223	013751	000000			PR0	; LEVEL 0
224	013752	000200			PR4	; LEVEL 4
225	013753	000240			PR5	; LEVEL 5
226	013754	000300			PR6	; LEVEL 6
227	013755	000340			PR7	; LEVEL 7
228	013756	105777	165520	INTTY:	TSTB	#STKS
229	013757	100375			BPL	.-4
230	013758	017703	165514		MOV	#STKB, R3
231	013759	105777	165512		TSTB	#STPS
232	013760	100375			BPL	.-4
233	013761	010377	165506		MOV	R3, #STPB
234	013762	042703	000240		BIC	#BIT7:BITS, R3
235	013763	000207			RTS	PC
236	013764			APT. SIZE:	RESET	
237	013765	000005			MOV	R0, -(SP)
238	013766	010046			MOV	R1, -(SP)
239	013767	010146			MOV	R2, -(SP)
240	013768	010246			MOV	R3, -(SP)
241	013769	010346			CLR	VECTR
242	013770	005037	013766		CLR	PRIRTY
243	013771	013700	001376		MOV	SCOM1, R0
244	013772	010037	001476		MOV	R0, SAVNUM
245	013773	012701	001346		MOV	#RANS1, R1
246	013774	013737	001372	013770	MOV	BASE, BASE
247	013775	113737	001366	013766	MOVB	SVECT1, VECTR
248	013776	113737	001367	013772	MOVB	SVECT1+1, PRIRTY
249	013777	013737	001374	001470	MOV	SDEVH, KRACTV
250	013778	013737	001470	001474	MOV	KRACTV, SAVACT
251	013779	012702	001402		MOV	#DDMO, R2
252	013780	012703	002100		MOV	#M1.MAP, R3
253	013781	005023		35:	CLR	(R3)+
254	013782	022703	002300		CHP	#M1.END, R3
255	013783	003374			BGT	35
256	013784	012703	002100		MOV	#M1.MAP, R3
257	013785	013723	013770	15:	MOV	BASE, (R3)+
258	013786	112163	000001		MOVB	(R1)+, 1(R3)
259	013787	006213			ASR	(R3)
260	013788	006213			ASR	(R3)
261	013789	053713	013772		BIS	PRIRTY, (R3)
262	013790	006313			RSL	(R3)
263	013791	006313			RSL	(R3)
264	013792	006313			RSL	(R3)
265	013793	006313			RSL	(R3)
266	013794	053723	013766		RIS	VECTR, (R3)+
267	013795	012223			MOV	(R2)+, (R3)+
268	013796	005723			TST	(R3)+

```

... PUSH R0 ON STACK
... PUSH R1 ON STACK
... PUSH R2 ON STACK
... PUSH R3 ON STACK
... CLEAR THE LOCAL VARIABLE
... CLEAN UP LOCAL VARIABLE
... GET THE DEVICE COUNT
... SAVE THE NO. OF DEVICES
... GET EXTRA INFO, BITS POINTER
... GET BASE CSR ADDRESS
... GET THE VECTOR
... GET THE PRIORITY
... SAVE THE KAC'S SELECTED ACTIVE
... SAVE THE ACTIVE REGISTER
... GET ADDRESS OF FIRST DEVICE DESCRIPTOR WORD
... GET POINTER TO DEVICE MAP
... CLEAR DEVICE MAP
... IS WHOLE DEV.MAP CLEARED?
... NO, THEN GO ON.
... RESTORE DEV.MAP POINTER.
... LOAD CSR ADDRESS
... GET EXTRA INFO. BITS
... SET IT IN RIGHT POSITION.
... SET IT IN RIGHT POSITION.
... GET PRIORITY IN STAT1
... SET THEM IN RIGHT POSITION
... " " " " " "
... " " " " " "
... GET THE VECTOR IN STAT1
... GET THE STAT2 FROM DC
... SKIP OVER STAT3
  
```

```

2244 013732 005300          DEC      RD          ; COUNT BY 1
2245 013734 001407          BEQ      ZS          ; ALL DONE?
2246 013736 062737 000010 013770  ADD      #10,BASE    ; INCREMENT BASE CSR ADDRESS BY 10
2247 013744 062737 000010 013766  ADD      #10,VECTR   ; INCREMENT VECTOR ADDRESS BY 10
2248 013752 000747          BR       IS          ; SET THE NEXT MAP ENTRY
2249 013754
2250 013754 012603          ZS:      MOV      (SP)+,R3    ; POP STACK INTO R3
2251 013756 012602          MOV      (SP)+,R2    ; POP STACK INTO R2
2252 013760 012601          MOV      (SP)+,R1    ; POP STACK INTO R1
2253 013762 012600          MOV      (SP)+,R0    ; POP STACK INTO R0
2254 013764 000207          RTS      PC          ; RETURN
2255 013766 000000          VECTR:  .WORD     0
2256 013770 000000          BASE:   .WORD     0
2257 013772 000000          PRTY:   .WORD     0
2258 013774          SFLT1   OUT1,INP1,4,0
2259 013774          SXZ
2260
2261 ;***** TEST 1 *****
2262 ;* MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
2263 ;* FLOAT A 1 THROUGH REGISTER OUT1 <4>
2264 ;* FLOAT A 0 THROUGH REGISTER OUT1 <4>
2265
2266 013774          SXZ
2267 ;*****
2268
2269 013774          STSTN
2270
2271 ; TEST 1
2272 013774 012737 000001 001202  TST1:  MOV      #1,STSTN    ; LOAD THE NO. OF THIS TEST
2273 014002 012737 014200 001442  MOV      #TST2,NEXT   ; POINT TO THE START OF NEXT TEST.
2274
2275 014010 004737 035536          JSR      PC,LDVRNT    ;R1 CONTAINS BASE KMC11 ADDRESS
2276 014014 014030          MCT1    ERROR      22 ;LOAD-VERIFY-WAIT.
2277 014016 104022          MOV      #STACK,SP   ;TIME OUT ERROR...
2278 014020 012706 001200          JMP      #NEXT       ;RESET STACK...
2279 014024 000177 165412          MCT1:   ;GO TO NEXT TEST...
2280 014030
2281 014030
2282 014030
2283          MOVE     #0,BREG    ;SET TO CLEAR SPAD 16
2284 014030 000400          MICPC=MICPC+1
2285 014032          .WORD     .S.
2286          MOVE     BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES...
2287          MICPC=MICPC+1
2288          .WORD     .S.
2289 014034          SFLT1   OUT1,INP1,4,0,1,15,25,35,45
2290 014034          IS:     MOVE     #200,BREG ;START WITH BIT 7.
2291          MICPC=MICPC+1
2292          .WORD     .S.
2293
2294          ZS:     MOVE     BREG,OUT1 <4> ;SET THE BIT.
2295          MICPC=MICPC+1
2296          .WORD     .S.
2297          MOVE     INP1 <4>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
2298          MICPC=MICPC+1
2299          .WORD     .S.
2299          MOVE     BREG,SPAD <4>

```

2300		000005	MICPC=MICPC+1	
2301	014042	063224	.WORD .S.	
2302	014044		SIFEQ BREG,SPAD <0> 3S	;CHECK THE DATA...
2303				
2304				
2305	014044		SUB2C SPAD <0>,BREG,NOP	
2306		000006	MICPC=MICPC+1	
2307	014044	060360	.WORD .S.	
2308	014046		BZ 3S	
2309		000007	MICPC=MICPC+1	
2310	014046	101422	.WORD .S.	
2311	014050		MOVE BREG,OUT1 <CSR4>	;GOOD DATA...
2312		000010	MICPC=MICPC+1	
2313	014050	061224	.WORD .S.	
2314	014052		MOVE IMP1 <4>,OUT1 <CSRS>;BAD DATA...	
2315		000011	MICPC=MICPC+1	
2316	014052	121105	.WORD .S.	
2317	014054		MOVE #1, MEM	;TYPE OF ERROR...
2318		000012	MICPC=MICPC+1	
2319	014054	002401	.WORD .S.	
2320	014056		MOVE MEM,OUT1 <CSR3>	;
2321		000013	MICPC=MICPC+1	
2322	014056	041223	.WORD .S.	
2323	014060		MOVE #4, MEM	;
2324		000014	MICPC=MICPC+1	
2325	014060	002404	.WORD .S.	
2326	014062		MOVE MEM,OUT1 <CSR7>	;REG. ADDRESS.
2327		000015	MICPC=MICPC+1	
2328	014062	041227	.WORD .S.	
2329	014064		CALL EROR	;REPORT DATA ERROR.
2330	014064		MOVE # <MICPC+3>,BREG	
2331		000016	MICPC=MICPC+1	
2332	014064	000420	.WORD .S.	
2333	014066		SBR EROR	
2334		000017	MICPC=MICPC+1	
2335	014066	104400	.WORD .S.	
2336	014070		MOVE SPAD <4>,BREG	;RESTORE BREG...
2337		000020	MICPC=MICPC+1	
2338	014070	060604	.WORD .S.	
2339	014072		SBR 2S	;LOOP ON ERROR...
2340		000021	MICPC=MICPC+1	
2341	014072	100403	.WORD .S.	
2342	014074		CALL SCP1	;IS LOOP DATA SET.
2343	014074		MOVE # <MICPC+3>,BREG	
2344		000022	MICPC=MICPC+1	
2345	014074	000424	.WORD .S.	
2346	014076		SBR SCP1	
2347		000023	MICPC=MICPC+1	
2348	014076	104427	.WORD .S.	
2349	014100		MOVE SPAD <4>,BREG	
2350		000024	MICPC=MICPC+1	
2351	014100	060604	.WORD .S.	
2352	014102		SBR 2S	;YES, DO IT.
2353		000025	MICPC=MICPC+1	
2354	014102	100403	.WORD .S.	
2355	014104		MOVE SPAD <4>,BREG	

3S:

2356 000026
 2357 014104 060604
 2358 014106
 2359 000027
 2360 014106 061620
 2361 014110
 2362 000030
 2363 014110 103432
 2364 014112
 2365 000031
 2366 014112 100403
 2367 014114
 2368 014114
 2369 014114
 2370 000032
 2371 014114 000577
 2372 014116
 2373 014116
 2374 000033
 2375 014116 061224
 2376 014120
 2377 000034
 2378 014120 123100
 2379 014122
 2380 000035
 2381 014122 063224
 2382 014124
 2383
 2384
 2385 014124
 2386 000036
 2387 014124 060360
 2388 014126
 2389 000037
 2390 014126 101452
 2391 014130
 2392 000040
 2393 014130 061224
 2394 014132
 2395 000041
 2396 014132 121105
 2397 014134
 2398 000042
 2399 014134 002401
 2400 014136
 2401 000043
 2402 014136 041223
 2403 014140
 2404 000044
 2405 014140 002404
 2406 014142
 2407 000045
 2408 014142 041227
 2409 014144
 2410 014144
 2411 000046

```

MICPC=MICPC+1
.WORD .S.
SHFBT ;NO, CONTINUE...
MICPC=MICPC+1
.WORD .SBR!..SELB!..DBRSH
BB7 45 ;IS IT DONE?...
MICPC=MICPC+1
.WORD .S.
SBR 25 ;NO, CONTINUE...
MICPC=MICPC+1
.WORD .S.
45:
SFLOT OUT1,INP1,4,0,0,115,125,135,145
115: MOVE #177,BREG ;START WITH BIT 7.
MICPC=MICPC+1
.WORD .S.
125:
MOVE BREG,OUT1 <4> ;SET THE BIT.
MICPC=MICPC+1
.WORD .S.
MOVE INP1 <4>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
MICPC=MICPC+1
.WORD .S.
MOVE BREG,SPAD <4>
MICPC=MICPC+1
.WORD .S.
SIFEQ BREG,SPAD <0> 135 ;CHECK THE DATA...

SUBPC SPAD <0>,BREG,NOP
MICPC=MICPC+1
.WORD .S.
BZ 135
MICPC=MICPC+1
.WORD .S.
MOVE BREG,OUT1 <CSR4> ;GOOD DATA...
MICPC=MICPC+1
.WORD .S.
MOVE INP1 <4>,OUT1 <CSR5>;BAD DATA...
MICPC=MICPC+1
.WORD .S.
MOVE #1,MEM ;TYPE OF ERROR...
MICPC=MICPC+1
.WORD .S.
MOVE MEM,OUT1 <CSR3> ;
MICPC=MICPC+1
.WORD .S.
MOVE #4,MEM ;
MICPC=MICPC+1
.WORD .S.
MOVE MEM,OUT1 <CSR7> ;REG. ADDRESS.
MICPC=MICPC+1
.WORD .S.
CALL EROR ;REPORT DATA ERROR.
MOVE # <MICPC+3>,BREG
MICPC=MICPC+1

```

014144 000450
014146 00047
014146 104400
014150 000050
014150 060604
014150 000051
014154 100433
014154 000052
014154 000454
014156 000053
014160 104427
014160 000054
014162 060604
014162 000055
014164 100433
014164 000056
014166 060604
014166 000057
014170 061620
014170 000060
014172 103433
014172
014172
014172
014172 000061
014174 000463
014174 000062
014174 104454
014176 000063
014176 100400
014200
014200
014200
014200

```
.WORD .S.  
SBR ERROR  
MICPC=MICPC+1  
.WORD .S.  
MOVE SPAD (4),BREG ;RESTORE BREG...  
MICPC=MICPC+1  
.WORD .S.  
SBR 128 ;LOOP ON ERROR...  
MICPC=MICPC+1  
135: .WORD .S.  
CALL SCP1 ;IS LOOP DATA SET.  
MOVE B (MICPC+3),BREG  
MICPC=MICPC+1  
.WORD .S.  
SBR SCP1  
MICPC=MICPC+1  
.WORD .S.  
MOVE SPAD (4),BREG  
MICPC=MICPC+1  
.WORD .S.  
SBR 128 ;YES, DO IT.  
MICPC=MICPC+1  
.WORD .S.  
MOVE SPAD (4),BREG  
MICPC=MICPC+1  
.WORD .S.  
SHFBRT ;NO, CONTINUE...  
MICPC=MICPC+1  
.WORD .SBR!.SELB!.DBRSH  
BB7 128 ;  
MICPC=MICPC+1  
145: .WORD .S.  
CALL SCPE  
MOVE B (MICPC+3),BREG  
MICPC=MICPC+1  
.WORD .S.  
SBR SCPE  
MICPC=MICPC+1  
.WORD .S.  
SBR 218  
MICPC=MICPC+1  
SFLT1 .WORD .S.  
SXZ OUT1,INP1,5,0  
  
;***** TEST 2 *****  
;* MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.  
;* FLOAT A 1 THROUGH REGISTER OUT1 (5)  
;* FLOAT A 0 THROUGH REGISTER OUT1 (5)  
SXZ ;*****  
STSTN ; TEST 2
```

014200	012737	000002	001202	TST2:	MOV	#2,STSTNM		; LOAD THE NO. OF THIS TEST
014206	012737	014404	001442		MOV	#TST3,NEXT		; POINT TO THE START OF NEXT TEST.
014214	004737	035536			JSR	PC,LDVANT		;R1 CONTAINS BASE KMC11 ADDRESS
014220	014234				MCT2			;LOAD-VERIFY-WAIT.
014222	104022				ERROR	22		; TIME OUT ERROR...
014224	012706	001200			MOV	#STACK,SP		;RESET STACK...
014230	000177	165206			JMP	#NEXT		;GO TO NEXT TEST...
014234				MCT2:				
014234				21S:	MOVE	#0,BREG		;SET TO CLEAR SPAD 16
014234	000000				MICPC=MICPC+1			
014236	000400				.WORD	.S.		
014236					MOVE	BREG,SPAD <16>		;FOR RETURN ADDRESS PURPOSES...
014236	000001				MICPC=MICPC+1			
014240	063236				.WORD	.S.		
014240				SFLOT	OUT1,INP1	#5,0,1,16,26,36,46		
014240				1S:	MOVE	#200,BREG		;START WITH BIT 7.
014240	000002				MICPC=MICPC+1			
014240	000600				.WORD	.S.		
014242				2S:	MOVE	BREG,OUT1 <5>		;SET THE BIT.
014242	000003				MICPC=MICPC+1			
014242	061225				.WORD	.S.		
014244					MOVE	INP1 <5>,SPAD <0>		;GET THE "FOUND" IN SCRATCH PAD.
014244	000004				MICPC=MICPC+1			
014244	123120				.WORD	.S.		
014246					MOVE	BREG,SPAD <4>		
014246	000005				MICPC=MICPC+1			
014246	063224				.WORD	.S.		
014250					SIFEQ	BREG,SPAD <0> 3S		;CHECK THE DATA...
014250					SUBPC	SPAD <0>,BREG,NOP		
014250	000006				MICPC=MICPC+1			
014250	060360				.WORD	.S.		
014252					BZ	3S		
014252	000007				MICPC=MICPC+1			
014252	101422				.WORD	.S.		
014254					MOVE	BREG,OUT1 <CSR4>		;GOOD DATA...
014254	000010				MICPC=MICPC+1			
014254	061224				.WORD	.S.		
014256					MOVE	INP1 <5>,OUT1 <CSR5>		;BAD DATA...
014256	000011				MICPC=MICPC+1			
014256	121125				.WORD	.S.		
014260					MOVE	#1,MEM		;TYPE OF ERROR...
014260	000012				MICPC=MICPC+1			
014260	002401				.WORD	.S.		
014262					MOVE	MEM,OUT1 <CSR3>		
014262	000013				MICPC=MICPC+1			
014262	041223				.WORD	.S.		
014264					MOVE	#5,MEM		
014264	000014				MICPC=MICPC+1			
014264	002405				.WORD	.S.		
014266					MOVE	MEM,OUT1 <CSR7>		;REG. ADDRESS.

014266 000015
014270 041227
014270 000016
014272 000420
014272 000017
014274 104400
014274 000020
014276 060604
014276 000021
014300 100403
014300 000022
014302 000424
014302 000023
014304 104427
014304 000024
014306 060604
014306 000025
014310 100403
014310 000026
014312 060604
014312 000027
014314 061620
014314 000030
014316 103432
014316 000031
014320 100403
014320 000032
014322 000577
014322 000033
014324 061225
014324 000034
014326 123120
014326 000035
014326 063224
014330

```
MICPC=MICPC+1  
.WORD .S.  
CALL EROR ;REPORT DATA ERROR.  
MOVE # <MICPC+3>,BREG  
MICPC=MICPC+1  
.WORD .S.  
SBR EROR  
MICPC=MICPC+1  
.WORD .S.  
MOVE SPAD <4>,BREG ;RESTORE BREG...  
MICPC=MICPC+1  
.WORD .S.  
SBR 25 ;LOOP ON ERROR...  
MICPC=MICPC+1  
35: CALL SCP1 ;IS LOOP DATA SET.  
MOVE # <MICPC+3>,BREG  
MICPC=MICPC+1  
.WORD .S.  
SBR SCP1  
MICPC=MICPC+1  
.WORD .S.  
MOVE SPAD <4>,BREG  
MICPC=MICPC+1  
.WORD .S.  
SBR 25 ;YES, DO IT.  
MICPC=MICPC+1  
MOVE SPAD <4>,BREG  
MICPC=MICPC+1  
.WORD .S.  
SHFBRT ;NO, CONTINUE...  
MICPC=MICPC+1  
.WORD .SBR! .SELB! .DBRSH  
BIT 45 ;IS IT DONE?...  
MICPC=MICPC+1  
.WORD .S.  
SBR 25 ;NO, CONTINUE...  
MICPC=MICPC+1  
45: .WORD .S.  
SFLOT OUT1,INP1,5,0,0,115,125,135,145  
115: MOVE # 177,BREG ;START WITH BIT 7.  
MICPC=MICPC+1  
.WORD .S.  
125: MOVE BREG,OUT1 <5> ;SET THE BIT.  
MICPC=MICPC+1  
.WORD .S.  
MOVE INP1 <5>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.  
MICPC=MICPC+1  
.WORD .S.  
MOVE BREG,SPAD <4>  
MICPC=MICPC+1  
.WORD .S.  
SIFEQ BREG,SPAD <0> 135 ;CHECK THE DATA...
```

2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635

014330 000036
014330 060360
014332 000037
014332 101452
014334 000040
014334 061224
014336 000041
014336 121125
014340 000042
014340 002401
014342 000043
014342 041223
014344 000044
014344 002405
014346 000045
014346 041227
014350 000046
014350 000450
014352 000047
014352 104400
014354 000050
014354 060604
014356 000051
014356 100433
014360 000052
014360 000454
014362 000053
014362 104427
014364 000054
014364 060604
014366 000055
014366 100433
014370 000056
014370 060604
014372

```
SUB2C SPAD <0>,BREG,NOP  
MICPC=MICPC+1  
.WORD .S.  
BZ 135  
MICPC=MICPC+1  
.WORD .S.  
MOVE BREG OUT1 <CSR4> ;GOOD DATA...  
MICPC=MICPC+1  
.WORD .S.  
MOVE INP1 <5>,OUT1 <CSRS>;BAD DATA...  
MICPC=MICPC+1  
.WORD .S.  
MOVE 0 1, MEM ;TYPE OF ERROR...  
MICPC=MICPC+1  
.WORD .S.  
MOVE MEM OUT1 <CSR3> ;  
MICPC=MICPC+1  
.WORD .S.  
MOVE 0 5, MEM ;  
MICPC=MICPC+1  
.WORD .S.  
MOVE MEM,OUT1 <CSR7> ;REG. ADDRESS.  
MICPC=MICPC+1  
.WORD .S.  
CALL EROR ;REPORT DATA ERROR.  
MOVE 0 <MICPC+3>,BREG  
MICPC=MICPC+1  
.WORD .S.  
SBR EROR  
MICPC=MICPC+1  
.WORD .S.  
MOVE SPAD <4>,BREG ;RESTORE BREG...  
MICPC=MICPC+1  
.WORD .S.  
SBR 125 ;LOOP ON ERROR...  
MICPC=MICPC+1  
.WORD .S.  
CALL SCP1 ;IS LOOP DATA SET.  
MOVE 0 <MICPC+3>,BREG  
MICPC=MICPC+1  
.WORD .S.  
SBR SCP1  
MICPC=MICPC+1  
.WORD .S.  
MOVE SPAD <4>,BREG  
MICPC=MICPC+1  
.WORD .S.  
SBR 125 ;YES, DO IT.  
MICPC=MICPC+1  
.WORD .S.  
MOVE SPAD <4>,BREG  
MICPC=MICPC+1  
.WORD .S.  
SHFBRT ;NO, CONTINUE...
```

135:

KMC11 MICRO PROCESSOR IBUS* TESTS

262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291

014372 000057
014374 061620
014374 000060
014376 103433
014376
014376
014376 000061
014400 000463
014400 000062
014402 104454
014402 000063
014404 100400
014404
014404
014404
014404 012737 000003 001202
014412 012737 014610 001442
014420 004737 035536
014424 014440
014426 104022
014430 012706 001200
014434 000177 165002
014440
014440
014440 000000
014442 000400
014442 000001
014444 063236
014444
014444
014444 000002
014446 000600
014446
014446
014446 000003
014450 061226
014450 000004

```
MICPC=MICPC+1
.WORD .S
SER! .SELB! .DBRSH
;
MICPC=MICPC+1
.WORD .S
145:
CALL SCPE
MOVE # <MICPC+3>, BREG
MICPC=MICPC+1
.WORD .S
SER SCPE
MICPC=MICPC+1
.WORD .S
SER 215
MICPC=MICPC+1
.WORD .S
SFLT1 OUT1, INP1, 6, 0
SXZ

;***** TEST 3 *****
;* MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
;* FLOAT A 1 THROUGH REGISTER OUT1 (6)
;* FLOAT A 0 THROUGH REGISTER OUT1 (6)
SXZ

;*****
STSTN
; TEST 3
-----
TST3: MOV #3, STSTNM ; LOAD THE NO. OF THIS TEST
MOV #STST4, NEXT ; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
;LOAD-VERIFY-WAIT.
JSR PC, LDVRMT
MCT3 ERROR 22 ;TIME OUT ERROR...
MOV #STACK, SP ;RESET STACK.
JMP #NEXT ;GO TO NEXT TEST...

215:
MOVE # 0, BREG ;SET TO CLEAR SPAD 16
MICPC=MICPC+1
.WORD .S
MOVE BREG, SPAD (16) ;FOR RETURN ADDRESS PURPOSES...
MICPC=MICPC+1
.WORD .S
SFLT0 OUT1, INP1, 6, 0, 1, 15, 25, 35, 45
15: MOVE # 200, BREG ;START WITH BIT 7.
MICPC=MICPC+1
.WORD .S
25:
MOVE BREG, OUT1 (6) ;SET THE BIT.
MICPC=MICPC+1
.WORD .S
MOVE INP1 (6), SPAD (0) ;GET THE "FOUND" IN SCRATCH PAD.
MICPC=MICPC+1
```

2698	014450	123140	WORD .S.	
2699	014452		MOVE BREG,SPAD <4>	
2700		000005	NICPC=NICPC+1	
2701	014452	063224	WORD .S.	
2702	014454		SIFEG BREG,SPAD <0> 3S	;CHECK THE DATA...
2703				
2704	014454	000006	SUB2C SPAD <0>,BREG,NOP	
2705	014454	060360	NICPC=NICPC+1	
2706	014456		WORD .S.	
2707	014456	000007	BZ 3S	
2708		101422	NICPC=NICPC+1	
2709	014456		WORD .S.	
2710	014460	000010	MOVE BREG,OUT1 <CSR4>	;GOOD DATA...
2711	014460	061224	NICPC=NICPC+1	
2712			WORD .S.	
2713	014462	000011	MOVE INP1 <6>,OUT1 <CSR5>;BAD DATA...	
2714	014462	121145	NICPC=NICPC+1	
2715	014464		WORD .S.	
2716	014464	000012	MOVE 8 I, MEM	;TYPE OF ERROR...
2717	014464	002401	NICPC=NICPC+1	
2718			WORD .S.	
2719	014466	000013	MOVE MEM,OUT1 <CSR3>	;
2720	014466	041223	NICPC=NICPC+1	
2721	014470		WORD .S.	
2722	014470	000014	MOVE 8 6 MEM	;
2723	014470	002406	NICPC=NICPC+1	
2724			WORD .S.	
2725	014472	000015	MOVE MEM,OUT1 <CSR7>	;REG. ADDRESS.
2726	014472	041227	NICPC=NICPC+1	
2727	014474		WORD .S.	
2728	014474		CALL EROR	;REPORT DATA ERROR.
2729	014474		MOVE 8 <NICPC+3>,BREG	
2730		000016	NICPC=NICPC+1	
2731	014474	000420	WORD .S.	
2732	014476		SBR EROR	
2733		000017	NICPC=NICPC+1	
2734	014476	104400	WORD .S.	
2735	014500		MOVE SPAD <4>,BREG	;RESTORE BREG...
2736	014500	000020	NICPC=NICPC+1	
2737	014502	060604	WORD .S.	
2738	014502		SBR 2S	;LOOP ON ERROR...
2739	014502	000021	NICPC=NICPC+1	
2740	014504	100403	WORD .S.	
2741	014504		CALL SCP1	;IS LOOP DATA SET.
2742	014504		MOVE 8 <NICPC+3>,BREG	
2743		000022	NICPC=NICPC+1	
2744	014504	000424	WORD .S.	
2745	014506		SBR SCP1	
2746	014506	000023	NICPC=NICPC+1	
2747	014510	104427	WORD .S.	
2748	014510		MOVE SPAD <4>,BREG	
2749	014510	000024	NICPC=NICPC+1	
2750	014510	060604	WORD .S.	
2751	014512		SBR 2S	;YES, DO IT.
2752		000025	NICPC=NICPC+1	

3S:

2748 014512 100403
 2749 014512 000026
 2750 014514 060604
 2751 014516 000027
 2752 014516 061620
 2753 014520 000030
 2754 014520 103432
 2755 014522 000031
 2756 014522 100403
 2757 014524 000032
 2758 014524 000577
 2759 014526 000033
 2760 014526 061226
 2761 014530 000034
 2762 014530 123140
 2763 014532 000035
 2764 014532 063224
 2765 014534 000036
 2766 014534 060360
 2767 014536 000037
 2768 014536 101452
 2769 014540 000040
 2770 014540 061224
 2771 014542 000041
 2772 014542 121145
 2773 014544 000042
 2774 014544 002401
 2775 014546 000043
 2776 014546 041223
 2777 014550 000044
 2778 014550 002406
 2779 014552 000045
 2780 014552 041227
 2781 014554

```

WORD .S.
MOVE SPAD <4>,BREG
MICPC=MICPC+1
WORD .S.
SHFRT ;NO, CONTINUE...
MICPC=MICPC+1
WORD .SBR!.SELB!.DBRSH
B87 45 ;IS IT DONE?...
MICPC=MICPC+1
WORD .S.
SBR 25 ;NO, CONTINUE...
MICPC=MICPC+1
WORD .S.
45:
SFLOT OUT1,INP1,6,0,0,115,125,135,145
115: MOVE #177,BREG ;START WITH BIT 7.
MICPC=MICPC+1
WORD .S.
125:
MOVE BREG,OUT1 <6> ;SET THE BIT.
MICPC=MICPC+1
WORD .S.
MOVE INP1 <6>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
MICPC=MICPC+1
WORD .S.
MOVE BREG,SPAD <4>
MICPC=MICPC+1
WORD .S.
SIFEQ BREG,SPAD <0> 135 ;CHECK THE DATA...

SUB2C SPAD <0>,BREG,NOP
MICPC=MICPC+1
WORD .S.
BZ 135
MICPC=MICPC+1
WORD .S.
MOVE BREG,OUT1 <CSR4> ;GOOD DATA...
MICPC=MICPC+1
WORD .S.
MOVE INP1 <6>,OUT1 <CSR5>;BAD DATA...
MICPC=MICPC+1
WORD .S.
MOVE #1,MEM ;TYPE OF ERROR...
MICPC=MICPC+1
WORD .S.
MOVE MEM,OUT1 <CSR3> ;
MICPC=MICPC+1
WORD .S.
MOVE #6,MEM ;
MICPC=MICPC+1
WORD .S.
MOVE MEM,OUT1 <CSR7> ;REG. ADDRESS.
MICPC=MICPC+1
WORD .S.
CALL EROR ;REPORT DATA ERROR.

```

2804 014554
2805 000046
2806 014554 000450
2807 014556
2808 000047
2809 014556 104400
2810 014560
2811 000050
2812 014560 060604
2813 014562
2814 000051
2815 014562 100433
2816 014564
2817 014564
2818 000052
2819 014564 000454
2820 014566
2821 000053
2822 014566 104427
2823 014570
2824 000054
2825 014570 060604
2826 014572
2827 000055
2828 014572 100433
2829 014574
2830 000056
2831 014574 060604
2832 014576
2833 000057
2834 014576 061620
2835 014600
2836 000060
2837 014600 103433
2838 014602
2839 014602
2840 014602
2841 000061
2842 014602 000463
2843 014604
2844 000062
2845 014604 104454
2846 014606
2847 000063
2848 014606 100400
2849 014610
2850 014610
2851 014610
2852 014610
2853 014610
2854 014610
2855 014610
2856 014610
2857 014610
2858 014610
2859 014610

```

MOVE      8 <MICPC+3>,BREG
MICPC=MICPC+1
.WORD     .S.
SBR       EROR
MICPC=MICPC+1
.WORD     .S.
MOVE      SPAD <4>,BREG           ;RESTORE BREG...
MICPC=MICPC+1
.WORD     .S.
SBR       128                     ;LOOP ON ERROR...
MICPC=MICPC+1
.WORD     .S.
CALL      SCP1                     ;IS LOOP DATA SET.
MOVE      8 <MICPC+3>,BREG
MICPC=MICPC+1
.WORD     .S.
SBR       SCP1
MICPC=MICPC+1
.WORD     .S.
MOVE      SPAD <4>,BREG
MICPC=MICPC+1
.WORD     .S.
SBR       128                     ;YES, DO IT.
MICPC=MICPC+1
.WORD     .S.
MOVE      SPAD <4>,BREG
MICPC=MICPC+1
.WORD     .S.
SHFBRT
MICPC=MICPC+1
.WORD     SBR!..SELB!..DBASH
BB7       128                     ;
MICPC=MICPC+1
.WORD     .S.
CALL      SCPE
MOVE      8 <MICPC+3>,BREG
MICPC=MICPC+1
.WORD     .S.
SBR       SCPE
MICPC=MICPC+1
.WORD     .S.
SBR       218
MICPC=MICPC+1
.WORD     .S.
SFLT1
SXZ      OUT1,INP1,7,0

;***** TEST 4 *****
;* MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
;* FLOAT A 1 THROUGH REGISTER OUT1 <?>
;* FLOAT A 0 THROUGH REGISTER OUT1 <?>
;*****
SXZ

```

KMC11 MICRO PROCESSOR IBUS* TESTS

```

014610 014610 012737 000004 001202 STSTN ; TEST 4
014616 012737 015014 001442 TST4: MOV #4,STSTNM ; LOAD THE NO. OF THIS TEST
; POINT TO THE START OF NEXT TEST.
; R1 CONTAINS BASE KMC11 ADDRESS
; LOAD-VERIFY-WAIT.
014624 004737 035536 JSR PC,LDVRMT
014630 014644 MCT4 ERROR 22 ; TIME OUT ERROR...
014636 012706 001200 MOV #STACK,SP ; RESET STACK...
014640 000177 164576 JMP #NEXT ; GO TO NEXT TEST...
014644 MCT4:
014644 215: MOVE #0,BREG ; SET TO CLEAR SPAD 16
MICPC=MICPC+1
.WORD .S.
014644 000000 MOVE BREG,SPAD <16> ; FOR RETURN ADDRESS PURPOSES...
.WORD .S.
014646 000400 MICPC=MICPC+1
.WORD .S.
014646 000001 SFLOT OUT1,INP1,7,0,1,15,25,35,45
15: MOVE #200,BREG ; START WITH BIT 7.
MICPC=MICPC+1
.WORD .S.
014650 000002 25: MOVE BREG,OUT1 <7> ; SET THE BIT.
MICPC=MICPC+1
.WORD .S.
014652 000600 MOVE INP1 <7>,SPAD <0> ; GET THE "FOUND" IN SCRATCH PAD.
MICPC=MICPC+1
.WORD .S.
014652 000003 MOVE BREG,SPAD <4>
MICPC=MICPC+1
.WORD .S.
014652 061227 SIFE0 BREG,SPAD <0> 35 ; CHECK THE DATA...
.WORD .S.
014654 000004 SUB2C SPAD <0>,BREG,NOP
MICPC=MICPC+1
.WORD .S.
014654 123160 BZ 35
MICPC=MICPC+1
.WORD .S.
014656 000005 MOVE BREG,OUT1 <CSR4> ; GOOD DATA...
MICPC=MICPC+1
.WORD .S.
014656 063224 MOVE INP1 <7>,OUT1 <CSR5>; BAD DATA...
MICPC=MICPC+1
.WORD .S.
014660 000006 MOVE #1,MEM ; TYPE OF ERROR...
MICPC=MICPC+1
.WORD .S.
014662 000007 MOVE MEM,OUT1 <CSR3> ;
MICPC=MICPC+1
.WORD .S.
014662 101422 MOVE #7,MEM ;
MICPC=MICPC+1
.WORD .S.
014664 000010
014664 061224
014666 000011
014666 121165
014670 000012
014670 002401
014672 000013
014672 041223
014674 000014

```


2916	014674	002407	.WORD .S.	
2917	014676		MOVE MEM,OUT1 <CSR7>	;REG. ADDRESS.
2918		000015	MICPC=MICPC+1	
2919	014676	041227	.WORD .S.	
2920	014700		CALL EROR	;REPORT DATA ERROR.
2921	014700		MOVE # <MICPC+3>,BREG	
2922		000016	MICPC=MICPC+1	
2923	014700	000420	.WORD .S.	
2924	014702		SBR EROR	
2925		000017	MICPC=MICPC+1	
2926	014702	104400	.WORD .S.	
2927	014704		MOVE SPAD <4>,BREG	;RESTORE BREG...
2928		000020	MICPC=MICPC+1	
2929	014704	060604	.WORD .S.	
2930	014706		SBR 25	;LOOP ON ERROR...
2931		000021	MICPC=MICPC+1	
2932	014706	100403	.WORD .S.	
2933	014710		CALL SCP1	;IS LOOP DATA SET.
2934	014710		MOVE # <MICPC+3>,BREG	
2935		000022	MICPC=MICPC+1	
2936	014710	000424	.WORD .S.	
2937	014712		SBR SCP1	
2938		000023	MICPC=MICPC+1	
2939	014712	104427	.WORD .S.	
2940	014714		MOVE SPAD <4>,BREG	
2941		000024	MICPC=MICPC+1	
2942	014714	060604	.WORD .S.	
2943	014716		SBR 25	;YES, DO IT.
2944		000025	MICPC=MICPC+1	
2945	014716	100403	.WORD .S.	
2946	014720		MOVE SPAD <4>,BREG	
2947		000026	MICPC=MICPC+1	
2948	014720	060604	.WORD .S.	
2949	014722		SFBRT	;NO, CONTINUE...
2950		000027	MICPC=MICPC+1	
2951	014722	061620	.WORD SBR!.SELB!.DBRSH	
2952	014724		B87 45	;IS IT DONE?...
2953		000030	MICPC=MICPC+1	
2954	014724	103432	.WORD .S.	
2955	014726		SBR 25	;NO, CONTINUE...
2956		000031	MICPC=MICPC+1	
2957	014726	100403	.WORD .S.	
2958	014730		45:	
2959	014730		SFLOT OUT1,INP1,7,0,0,115,125,135,145	
2960	014730		115: MOVE # 177,BREG ;START WITH BIT 7.	
2961		000032	MICPC=MICPC+1	
2962	014730	000577	.WORD .S.	
2963	014732		125:	
2964	014732		MOVE BREG,OUT1 <7> ;SET THE BIT.	
2965		000033	MICPC=MICPC+1	
2966	014732	061227	.WORD .S.	
2967	014734		MOVE INP1 <7>,SPAD <0>	;GET THE "FOUND" IN SCRATCH PAD.
2968		000034	MICPC=MICPC+1	
2969	014734	123160	.WORD .S.	
2970	014736		MOVE BREG,SPAD <4>	
2971		000035	MICPC=MICPC+1	

2972	014736	063224	.WORD	.S.		
2973	014740		SIFED	BREG,SPAD <0>	135	;CHECK THE DATA...
2974						
2975	014740		SUBFC	SPAD <0>,BREG,NOP		
2976	014740	000036	NICPC=NICPC+1			
2977	014742	060360	.WORD	.S.		
2978	014742		BZ	135		
2979	014742	000037	NICPC=NICPC+1			
2980	014742	101462	.WORD	.S.		
2981	014744		MOVE	BREG,OUT1 <CSR4>		;GOOD DATA...
2982	014744	000040	NICPC=NICPC+1			
2983	014744	061224	.WORD	.S.		
2984	014746		MOVE	INP1 <7>,OUT1 <CSR5>		;BAD DATA...
2985	014746	000041	NICPC=NICPC+1			
2986	014746	121166	.WORD	.S.		
2987	014750		MOVE	8,1,MEH		;TYPE OF ERROR...
2988	014750	000042	NICPC=NICPC+1			
2989	014752	002401	.WORD	.S.		
2990	014752		MOVE	MEH,OUT1 <CSR3>		;
2991	014752	000043	NICPC=NICPC+1			
2992	014754	041223	.WORD	.S.		
2993	014754		MOVE	8,7,MEH		;
2994	014754	000044	NICPC=NICPC+1			
2995	014754	002407	.WORD	.S.		
2996	014756		MOVE	MEH,OUT1 <CSR7>		;REG. ADDRESS.
2997	014756	000045	NICPC=NICPC+1			
2998	014756	041227	.WORD	.S.		
2999	014760		CALL	ENOR		;REPORT DATA ERROR.
3000	014760		MOVE	8 <NICPC+3>,BREG		
3001	014760	000046	NICPC=NICPC+1			
3002	014762	000450	.WORD	.S.		
3003	014762		SBR	ENOR		
3004	014762	000047	NICPC=NICPC+1			
3005	014764	104400	.WORD	.S.		
3006	014764		MOVE	SPAD <4>,BREG		;RESTORE BREG...
3007	014764	000050	NICPC=NICPC+1			
3008	014764	060604	.WORD	.S.		
3009	014766		SBR	125		;LOOP ON ERROR...
3010	014766	000051	NICPC=NICPC+1			
3011	014766	100433	.WORD	.S.		
3012	014770		CALL	SCP1		;IS LOOP DATA SET.
3013	014770		MOVE	8 <NICPC+3>,BREG		
3014	014770	000052	NICPC=NICPC+1			
3015	014772	000454	.WORD	.S.		
3016	014772		SBR	SCP1		
3017	014772	000053	NICPC=NICPC+1			
3018	014772	104427	.WORD	.S.		
3019	014774		MOVE	SPAD <4>,BREG		
3020	014774	000054	NICPC=NICPC+1			
3021	014776	060604	.WORD	.S.		
3022	014776		SBR	125		;YES, DO IT.
3023	014776	000055	NICPC=NICPC+1			
3024	014776	100433	.WORD	.S.		
3025	015000		MOVE	SPAD <4>,BREG		
3026	015000	000056	NICPC=NICPC+1			
3027						

135:

27(1006) 13-MAY-77 14:07 PAGE 61
13-MAY-77 13-58

KMC11 MICRO PROCESSOR IBUS* TESTS

```

015000 060604 .WORD .S.
015002 SHFBRT ;NO, CONTINUE...
000057 MICPC=MICPC+1
015002 061620 .WORD SBR!.SELB!.DBRSH
015004 BB7 128 ;
000060 MICPC=MICPC+1
015004 103433 .WORD .S.
145: CALL SCPE
MOVE # (MICPC+3),BREG
MICPC=MICPC+1
015006 000061 .WORD .S.
015006 000463 SBR SCPE
015010 000062 MICPC=MICPC+1
015010 104454 .WORD .S.
015012 000063 SBR 215
015012 100400 MICPC=MICPC+1
SFLT1 .WORD .S.
SXZ OUT1,INP1,10,236

;***** TEST 5 *****
; MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
; FLOAT A 1 THROUGH REGISTER OUT1 <10>
; FLOAT A 0 THROUGH REGISTER OUT1 <10>
; THE NPR R0 BIT (BIT0) IS MASKED DURING THIS TEST.
3055 015014 SXZ
;*****

3058 015014 STSTN
; TEST 5
-----
3061 015014 012737 000005 001202 TST5: MOV #5,STSTNM ; LOAD THE NO. OF THIS TEST
3062 015022 012737 015234 001442 MOV #STST6,NEXT ; POINT TO THE START OF NEXT TEST.
3064 015030 004737 035536 JSR PC,LDVRMT ;R1 CONTAINS BASE KMC11 ADDRESS
3065 015034 015050 MCT5 ;LOAD-VERIFY-WAIT.
3066 015036 104022 ERROR 22 ;TIME OUT ERROR...
3067 015040 012706 001200 MOV #STACK,SP ;RESET STACK...
3068 015044 000177 164372 JMP #NEXT ;GO TO NEXT TEST...
MCT5:
215: MOVE #0,BREG ;SET TO CLEAR SPAD 16
MICPC=MICPC+1
015050 000000 .WORD .S.
015052 000400 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES...
MICPC=MICPC+1
015052 063236 .WORD .S.
SFL0T OUT1,INP1,10,236,1,15,25,35,45
15: MOVE #200,BREG ;START WITH BIT 7.
MICPC=MICPC+1
015054 000002 .WORD .S.
015056 000600
25: MOVE BREG,SPAD <4> ;SAVE BREGISTER.
MICPC=MICPC+1
015056 000003

```

3084	015056	063224	.WORD .S.	
3085	015060		MOVE # 236, MEM	; CLEAR THE UNWANTED BITS
3086		000004	MICPC=MICPC+1	
3087	015060	002636	.WORD .S.	
3088	015062		MOVE MEM SPAD <1>	; CLEAR THE UNWANTED BITS.
3089		000005	MICPC=MICPC+1	
3090	015062	043221	.WORD .S.	
3091	015064		AND SPAD <1>, BREG, BREG	; CLEAR THE UNWANTED BITS.
3092		000006	MICPC=MICPC+1	
3093	015064	060661	.WORD .S.	
3094	015066		MOVE BREG, OUT1 <10>	; SET THE BIT.
3095		000007	MICPC=MICPC+1	
3096	015066	061230	.WORD .S.	
3097	015070		MOVE INP1 <10>, SPAD <0>	; GET THE "FOUND" IN SCRATCH PAD.
3098		000010	MICPC=MICPC+1	
3099	015070	123200	.WORD .S.	
3100	015072		\$IFEQ BREG, SPAD <0> 3S	; CHECK THE DATA...
3101				
3102				
3103	015072		SUB2C SPAD <0>, BREG, NCP	
3104		000011	MICPC=MICPC+1	
3105	015072	060360	.WORD .S.	
3106	015074		BZ 3S	
3107		000012	MICPC=MICPC+1	
3108	015074	101425	.WORD .S.	
3109	015076		MOVE BREG, OUT1 <CSR4>	; GOOD DATA...
3110		000013	MICPC=MICPC+1	
3111	015076	061224	.WORD .S.	
3112	015100		MOVE INP1 <10>, OUT1 <CSR5>	; BAD DATA...
3113		000014	MICPC=MICPC+1	
3114	015100	121205	.WORD .S.	
3115	015102		MOVE # 1, MEM	; TYPE OF ERROR...
3116		000015	MICPC=MICPC+1	
3117	015102	002401	.WORD .S.	
3118	015104		MOVE MEM, OUT1 <CSR3>	;
3119		000016	MICPC=MICPC+1	
3120	015104	041223	.WORD .S.	
3121	015106		MOVE # 10, MEM	;
3122		000017	MICPC=MICPC+1	
3123	015106	002410	.WORD .S.	
3124	015110		MOVE MEM, OUT1 <CSR7>	; REG. ADDRESS.
3125		000020	MICPC=MICPC+1	
3126	015110	041227	.WORD .S.	
3127	015112		CALL EROR	; REPORT DATA ERROR.
3128	015112		MOVE # <MICPC+3>, BREG	
3129		000021	MICPC=MICPC+1	
3130	015112	000423	.WORD .S.	
3131	015114		SBR EROR	
3132		000022	MICPC=MICPC+1	
3133	015114	104400	.WORD .S.	
3134	015116		MOVE SPAD <4>, BREG	; RESTORE BREG...
3135		000023	MICPC=MICPC+1	
3136	015116	060604	.WORD .S.	
3137	015120		SBR 2S	; LOOP ON ERROR...
3138		000024	MICPC=MICPC+1	
3139	015120	100403	.WORD .S.	

3140	015122		35:	CALL	SCP1		; IS LOOP DATA SET.
3141	015122			MOVE	# (MICPC+3), BREG		
3142		000025		MICPC=MICPC+1			
3143	015122	000427		.WORD	.S.		
3144	015124			SBR	SCP1		
3145		000026		MICPC=MICPC+1			
3146	015124	104427		.WORD	.S.		
3147	015126			MOVE	SPAD (4), BREG		
3148		000027		MICPC=MICPC+1			
3149	015126	060604		.WORD	.S.		
3150	015130			SBR	25		; YES, DO IT.
3151		000030		MICPC=MICPC+1			
3152	015130	100403		.WORD	.S.		
3153	015132			MOVE	SPAD (4), BREG		
3154		000031		MICPC=MICPC+1			
3155	015132	060604		.WORD	.S.		
3156	015134			SHFBRT			; NO, CONTINUE...
3157		000032		MICPC=MICPC+1			
3158	015134	061620		.WORD	SBR!.SELB!.DBRSH		
3159	015136			B97	45		; IS IT DONE?...
3160		000033		MICPC=MICPC+1			
3161	015136	103435		.WORD	.S.		
3162	015140			SBR	25		; NO, CONTINUE...
3163		000034		MICPC=MICPC+1			
3164	015140	100403		.WORD	.S.		
3165	015142		45:				
3166	015142		SFLOT	OUT1, INP1, 10, 236, 0, 115, 125, 135, 145			
3167	015142		115:	MOVE	# 177, BREG ; START WITH BIT 7.		
3168		000035		MICPC=MICPC+1			
3169	015142	000577		.WORD	.S.		
3170	015144		125:				
3171	015144			MOVE	BREG, SPAD (4)		; SAVE BREGISTER.
3172		000036		MICPC=MICPC+1			
3173	015144	063224		.WORD	.S.		
3174	015146			MOVE	# 236, MEM		; CLEAR THE UNWANTED BITS
3175		000037		MICPC=MICPC+1			
3176	015146	002636		.WORD	.S.		
3177	015150			MOVE	MEM, SPAD (1)		; CLEAR THE UNWANTED BITS.
3178		000040		MICPC=MICPC+1			
3179	015150	043221		.WORD	.S.		
3180	015152			AND	SPAD (1), BREG, BREG		; CLEAR THE UNWANTED BITS.
3181		000041		MICPC=MICPC+1			
3182	015152	060661		.WORD	.S.		
3183	015154			MOVE	BREG, OUT1 (10) ; SET THE BIT.		
3184		000042		MICPC=MICPC+1			
3185	015154	061230		.WORD	.S.		
3186	015156			MOVE	INP1 (10), SPAD (0)		; GET THE "FOUND" IN SCRATCH PAD.
3187		000043		MICPC=MICPC+1			
3188	015156	123200		.WORD	.S.		
3189	015160			SIFEQ	BREG, SPAD (0) 135		; CHECK THE DATA...
3190							
3191							
3192	015160			SUB2C	SPAD (0), BREG, NOP		
3193		000044		MICPC=MICPC+1			
3194	015160	060360		.WORD	.S.		
3195	015162			BZ	135		

3196		000045	MICPC=MICPC+1	
3197	015162	101460	.WORD .S	
3198	015164		MOVE BREG,OUT1 <CSR4>	;GOOD DATA...
3199		000046	MICPC=MICPC+1	
3200	015164	061224	.WORD .S	
3201	015166		MOVE INP1 <10>,OUT1 <CSRS>;	BAD DATA...
3202		000047	MICPC=MICPC+1	
3203	015166	121205	.WORD .S	
3204	015170		MOVE #1, MEM	;TYPE OF ERROR...
3205		000050	MICPC=MICPC+1	
3206	015170	002401	.WORD .S	
3207	015172		MOVE MEM,OUT1 <CSR3>	;
3208		000051	MICPC=MICPC+1	
3209	015172	041223	.WORD .S	
3210	015174		MOVE #10, MEM	;
3211		000052	MICPC=MICPC+1	
3212	015174	002410	.WORD .S	
3213	015176		MOVE MEM,OUT1 <CSR7>	;REG. ADDRESS.
3214		000053	MICPC=MICPC+1	
3215	015176	041227	.WORD .S	
3216	015200		CALL EROR	;REPORT DATA ERROR.
3217	015200		MOVE # <MICPC+3>,BREG	
3218		000054	MICPC=MICPC+1	
3219	015200	000456	.WORD .S	
3220	015202		SBR EROR	
3221		000055	MICPC=MICPC+1	
3222	015202	104400	.WORD .S	
3223	015204		MOVE SPAD <4>,BREG	;RESTORE BREG...
3224		000056	MICPC=MICPC+1	
3225	015204	060604	.WORD .S	
3226	015206		SBR 125	;LOOP ON ERROR...
3227		000057	MICPC=MICPC+1	
3228	015206	100436	.WORD .S	
3229	015210		CALL SCP1	;IS LOOP DATA SET.
3230	015210		MOVE # <MICPC+3>,BREG	
3231		000060	MICPC=MICPC+1	
3232	015210	000462	.WORD .S	
3233	015212		SBR SCP1	
3234		000061	MICPC=MICPC+1	
3235	015212	104427	.WORD .S	
3236	015214		MOVE SPAD <4>,BREG	
3237		000062	MICPC=MICPC+1	
3238	015214	060604	.WORD .S	
3239	015216		SBR 125	;YES, DO IT.
3240		000063	MICPC=MICPC+1	
3241	015216	100436	.WORD .S	
3242	015220		MOVE SPAD <4>,BREG	
3243		000064	MICPC=MICPC+1	
3244	015220	060604	.WORD .S	
3245	015222		SFBRT	;NO, CONTINUE...
3246		000065	MICPC=MICPC+1	
3247	015222	061620	.WORD SBR!.SELB!.DBRSH	
3248	015224		BB7 125	;
3249		000066	MICPC=MICPC+1	
3250	015224	103436	.WORD .S	
3251	015226			

135:

145:

KMC11 MICRO PROCESSOR IBUS* TESTS

015226
015226
015226
015230
015230
015232
015232
015234
015234
015234
015234
015234
015242
015250
015254
015256
015260
015270
015270
015270
015270
015270
015270
015272
015274
015274
015274
015274
015276
015276
015276
015276
015300
015300
015302
015302
015302
3307

```

CALL SCPE
MOVE # <MICPC+3>, BREG
MICPC=MICPC+1
.WORD .S.
SBR SCPE
MICPC=MICPC+1
.WORD .S.
SBR 21$
MICPC=MICPC+1
.WORD .S.
SFLT1 OUT1, INP1, 11, 015
SXZ

;***** TEST 6 *****
; MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
; FLOAT A 1 THROUGH REGISTER OUT1 <11>
; FLOAT A 0 THROUGH REGISTER OUT1 <11>
; THE BR AQ BIT, PGM CLOCK BIT, FORCE POWER FAIL BIT
; (BITS 7,4,1) ARE ALL MASKED DURING THIS TEST

;*****
; TEST 6
-----
TST6: MOV #6, STSTNM ; LOAD THE NO. OF THIS TEST
MOV #TST7, NEXT ; POINT TO THE START OF NEXT TEST.
; R1 CONTAINS BASE KMC11 ADDRESS
; LOAD-VERIFY-WAIT.
JSR PC, LDVRMT
MCT6 ERROR 22 ; TIME OUT ERROR...
MOV #STACK, SP ; RESET STACK...
JMP @NEXT ; GO TO NEXT TEST...

MCT6: 21$: MOVE # 0, BREG ; SET TO CLEAR SPAD 16
MICPC=MICPC+1
.WORD .S.
MOVE BREG, SPAD <16> ; FOR RETURN ADDRESS PURPOSES...
MICPC=MICPC+1
.WORD .S.
SFLOT OUT1, INP1, 11, 015, 1, 15, 25, 35, 45
1$: MOVE # 200, BREG ; START WITH BIT 7.
MICPC=MICPC+1
.WORD .S.

2$: MOVE BREG, SPAD <4> ; SAVE BREGISTER.
MICPC=MICPC+1
.WORD .S.
MOVE # 015, MEM ; CLEAR THE UNWANTED BITS
MICPC=MICPC+1
.WORD .S.
MOVE MEM, SPAD <1> ; CLEAR THE UNWANTED BITS.
MICPC=MICPC+1
.WORD .S.

```


3308	015304	000006	AND SPAD <1>, BREG, BREG ;CLEAR THE UNWANTED BITS.
3309			MICPC=MICPC+1
3310	015304	060661	.WORD .S.
3311	015306		MOVE BREG, OUT1 <11> ;SET THE BIT.
3312		000007	MICPC=MICPC+1
3313	015306	061231	.WORD .S.
3314	015310		MOVE INP1 <11>, SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
3315		000010	MICPC=MICPC+1
3316	015310	123220	.WORD .S.
3317	015312		MOVE # 20, MEM ;SET TO SET PGM.CLK BIT.
3318		000011	MICPC=MICPC+1
3319	015312	002420	.WORD .S.
3320	015314		MOVE MEM, SPAD <1> ; " " " " " " " " " " " "
3321		000012	MICPC=MICPC+1
3322	015314	043221	.WORD .S.
3323	015316		OR SPAD <1>, BREG, BREG ;SET PGM.CLK. BIT IN BREG.
3324		000013	MICPC=MICPC+1
3325	015316	060701	.WORD .S.
3326	015320		SIFE0 BREG, SPAD <0> 3S ;CHECK THE DATA...
3327			
3328	015320		
3329		000014	SUBC SPAD <0>, BREG, NOP
3330			MICPC=MICPC+1
3331	015320	060360	.WORD .S.
3332	015322		BZ 3S
3333		000015	MICPC=MICPC+1
3334	015322	101430	.WORD .S.
3335	015324		MOVE BREG, OUT1 <CSR4> ;GOOD DATA...
3336		000016	MICPC=MICPC+1
3337	015324	061224	.WORD .S.
3338	015326		MOVE INP1 <11>, OUT1 <CSRS>;BAD DATA...
3339		000017	MICPC=MICPC+1
3340	015326	121225	.WORD .S.
3341	015330		MOVE # 1, MEM ;TYPE OF ERROR...
3342		000020	MICPC=MICPC+1
3343	015330	002401	.WORD .S.
3344	015332		MOVE MEM, OUT1 <CSR3> ;
3345		000021	MICPC=MICPC+1
3346	015332	041223	.WORD .S.
3347	015334		MOVE # 11, MEM ;
3348		000022	MICPC=MICPC+1
3349	015334	002411	.WORD .S.
3350	015336		MOVE MEM, OUT1 <CSR7> ;REG. ADDRESS.
3351		000023	MICPC=MICPC+1
3352	015336	041227	.WORD .S.
3353	015340		CALL EROR ;REPORT DATA ERROR.
3354	015340		MOVE # <MICPC+3>, BREG
3355		000024	MICPC=MICPC+1
3356	015340	000426	.WORD .S.
3357	015342		SBR EROR
3358		000025	MICPC=MICPC+1
3359	015342	104400	.WORD .S.
3360	015344		MOVE SPAD <4>, BREG ;RESTORE BREG...
3361		000026	MICPC=MICPC+1
3362	015344	060604	.WORD .S.
3363	015346		SBR 2S ;LOOP ON ERROR...

1-2

```

3364 000027      NICPC=NICPC+1
3365 015376 100403      .WORD      .S.
3366 015378 100403      35:  CALL      SCP1      ;IS LOOP DATA SET.
3367 015380 100403      MOVE      B <NICPC+3>,BREG
3368 000030      NICPC=NICPC+1
3369 015380 000432      .WORD      .S.
3370 015382 100403      SBR      SCP1
3371 000031      NICPC=NICPC+1
3372 015382 104427      .WORD      .S.
3373 015384 104427      MOVE      SPAD <4>,BREG
3374 000032      NICPC=NICPC+1
3375 015384 060604      .WORD      .S.
3376 015386 060604      SBR      25      ;YES, DO IT.
3377 000033      NICPC=NICPC+1
3378 015386 100403      .WORD      .S.
3379 015388 100403      MOVE      SPAD <4>,BREG
3380 000034      NICPC=NICPC+1
3381 015388 060604      .WORD      .S.
3382 015382 000034      SHFBRT      ;NO, CONTINUE...
3383 015382 000035      NICPC=NICPC+1
3384 015384 061620      .WORD      SBR!..SELB!..DBRSH
3385 015384 061620      BB7      45      ;IS IT DONE?...
3386 015384 000036      NICPC=NICPC+1
3387 015386 103440      .WORD      .S.
3388 015386 000037      SBR      25      ;NO, CONTINUE...
3389 015386 000037      NICPC=NICPC+1
3390 015386 100403      .WORD      .S.
3391 015370 45: SFLOT OUT1,INP1,11,015,0,115,125,135,145
3392 015370 115: MOVE B 177,BREG ;START WITH BIT 7.
3393 015370 000040      NICPC=NICPC+1
3394 015370 000577      .WORD      .S.
3395 015372 125: MOVE BREG,SPAD <4>      ;SAVE BREGISTER.
3396 015372 000041      NICPC=NICPC+1
3397 015372 063224      .WORD      .S.
3398 015374 000042      MOVE      B 015,MEM      ;CLEAR THE UNWANTED BITS
3399 015374 002415      NICPC=NICPC+1
3400 015374 002415      .WORD      .S.
3401 015376 000043      MOVE      MEM,SPAD <1>      ;CLEAR THE UNWANTED BITS.
3402 015376 043221      NICPC=NICPC+1
3403 015400 000044      .WORD      .S.
3404 015400 060661      AND      SPAD <1>,BREG,BREG      ;CLEAR THE UNWANTED BITS.
3405 015402 000045      NICPC=NICPC+1
3406 015402 061231      MOVE      BREG,OUT1 <11> ;SET THE BIT.
3407 015404 000046      .WORD      .S.
3408 015404 123220      MOVE      INP1 <11>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
3409 015406 000047      NICPC=NICPC+1
3410 015406 002420      .WORD      .S.
3411 015406 000047      MOVE      B 20,MEM      ;SET TO SET PGM.CLK BIT.
3412 015410 000048      NICPC=NICPC+1
3413 015410 000048      .WORD      .S.
3414 015410 000048      MOVE      MEM,SPAD <1>      ; " " " " " "
3415 015410 000048      NICPC=NICPC+1

```

3420	015410	043221	.WORD	.S.		
3421	015412		OR	SPAD <1>	,BREG,BREG	;SET PGM.CLK. BIT IN BREG.
		000051	MICPC=MICPC+1			
	015412	060701	.WORD	.S.		
	015414		SIFEQ	BREG,SPAD <0>	13S	;CHECK THE DATA...
	015414		SUB2C	SPAD <0>	,BREG,NOP	
		000052	MICPC=MICPC+1			
	015414	060360	.WORD	.S.		
	015416		BZ	13S		
		000053	MICPC=MICPC+1			
	015416	101466	.WORD	.S.		
	015420		MOVE	BREG,OUT1 <CSR4>		;GOOD DATA...
		000054	MICPC=MICPC+1			
	015420	061224	.WORD	.S.		
	015422		MOVE	INP1 <11>	,OUT1 <CSRS>	;BAD DATA...
		000055	MICPC=MICPC+1			
	015422	121225	.WORD	.S.		
	015424		MOVE	# 1, MEM		;TYPE OF ERROR...
		000056	MICPC=MICPC+1			
	015424	002401	.WORD	.S.		
	015426		MOVE	MEM,OUT1 <CSR3>		
		000057	MICPC=MICPC+1			
	015426	041223	.WORD	.S.		
	015430		MOVE	# 11, MEM		
		000060	MICPC=MICPC+1			
	015430	002411	.WORD	.S.		
	015432		MOVE	MEM,OUT1 <CSR7>		;REG. ADDRESS.
		000061	MICPC=MICPC+1			
	015432	041227	.WORD	.S.		
	015434		CALL	ERROR		;REPORT DATA ERROR.
		000062	MOVE	# <MICPC+3>	,BREG	
		000464	MICPC=MICPC+1			
	015434		.WORD	.S.		
	015436		SBR	ERROR		
		000063	MICPC=MICPC+1			
	015436	104400	.WORD	.S.		
	015440		MOVE	SPAD <4>	,BREG	;RESTORE BREG...
		000064	MICPC=MICPC+1			
	015440	060604	.WORD	.S.		
	015442		SBR	12S		;LOOP ON ERROR...
		000065	MICPC=MICPC+1			
	015442	100441	.WORD	.S.		
	015444		CALL	SCP1		;IS LOOP DATA SET.
		000066	MOVE	# <MICPC+3>	,BREG	
		000470	MICPC=MICPC+1			
	015444		.WORD	.S.		
	015446		SBR	SCP1		
		000067	MICPC=MICPC+1			
	015446	104427	.WORD	.S.		
3470	015446		MOVE	SPAD <4>	,BREG	
3471	015450		MICPC=MICPC+1			
3472		000070	.WORD	.S.		
3473	015450	060604	SBR	12S		;YES, DO IT.
3474	015452		MICPC=MICPC+1			
3475		000071				

13S:

KMC11 MICRO PROCESSOR IBUS* TESTS

3476 015452 100441
 3477 015454
 3478 000072
 3479 015454 060604
 3480 015456
 3481 000073
 3482 015456 061620
 3483 015460
 3484 000074
 3485 015460 103441
 3486 015462
 3487 015462
 3488 000075
 3489 015462 000477
 3490 015464
 3491 000076
 3492 015464 104454
 3493 015466
 3494 000077
 3495 015466 100400
 3496 015470
 3497 015470
 3498 015470
 3499 015470
 3500 015470
 3501 015470
 3502 015470
 3503 015470
 3504 015470
 3505 015470
 3506 015470
 3507 015470
 3508 015470
 3509 015470
 3510 015470
 3511 015470 012737 000007 001202
 3512 015476 012737 015674 001442
 3513 015504 004737 035536
 3514 015510 015524
 3515 015512 104022
 3516 015514 012706 001200
 3517 015520 000177 163716
 3518 015524
 3519 015524 000000
 3520 015526 000400
 3521 015526 000001
 3522 015526 063236
 3523 015530
 3524 015530
 3525 015530 000002
 3526 015530 000600
 3527 015532
 3528 015532
 3529 015532
 3530 015532
 3531 015532

```

WORD .S
MOVE SPAD <4>,BREG
MICPC=MICPC+1
WORD .S
SFBRT ;NO, CONTINUE...
MICPC=MICPC+1
WORD SBR!.SELB!.DBRSH
;
BRT 125
MICPC=MICPC+1
WORD .S
145:
CALL SCPE
MOVE # <MICPC+3>,BREG
MICPC=MICPC+1
WORD .S
SBR SCPE
MICPC=MICPC+1
WORD .S
SBR 215
MICPC=MICPC+1
WORD .S
SFLT1 OUTO,INPO,0,0
SXZ

;***** TEST 7 *****
;* MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
;* FLOAT A 1 THROUGH REGISTER OUTO <0>
;* FLOAT A 0 THROUGH REGISTER OUTO <0>
SXZ

;*****
STSTN
; TEST 7
-----
TST7: MOV #7,STSTNM ; LOAD THE NO. OF THIS TEST
MOV #TST10,NEXT ; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
JSR PC,LVWRM ;LOAD-VERIFY-WAIT.
MCT7 ERROR 22 ;TIME OUT ERROR...
MOV #STACK,SP ;RESET STACK...
JMP #NEXT ;GO TO NEXT TEST...

MCT7:
215:
MOVE # 0,BREG ;SET TO CLEAR SPAD 16
MICPC=MICPC+1
WORD .S
MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES...
MICPC=MICPC+1
WORD .S
SFLOT OUTO,INPO,0,0,1,15,25,35,45
15: MOVE # 200,BREG ;START WITH BIT 7.
MICPC=MICPC+1
WORD .S
25:

```

KMC11 MICRO PROCESSOR IBUS TESTS

```

353 015532 000003 MOVE BREG,OUT0 <0> ;SET THE BIT.
015532 062220 MICPC=MICPC+1
015534 000004 .WORD .S.
015534 023000 MOVE INPO <0>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
015536 000005 MICPC=MICPC+1
015536 063224 .WORD .S.
015540 000006 MOVE BREG,SPAD <4>
015540 063224 MICPC=MICPC+1
015540 000007 SIFEQ BREG,SPAD <0> 35 ;CHECK THE DATA...
015540 000008 SUBRC SPAD <0>,BREG,NOP
015540 060360 MICPC=MICPC+1
015542 000009 .WORD .S.
015542 101422 BZ 35
015544 000010 MICPC=MICPC+1
015544 061224 MOVE BREG,OUT1 <CSR4> ;GOOD DATA...
015546 000011 MICPC=MICPC+1
015546 021005 MOVE INPO <0>,OUT1 <CSR5>;BAD DATA...
015548 000012 .WORD .S.
015550 000013 MOVE B I MEM ;TYPE OF ERROR...
015550 002401 MICPC=MICPC+1
015552 000014 .WORD .S.
015554 000015 MOVE MEM,OUT1 <CSR3> ;
015554 002400 MICPC=MICPC+1
015556 000016 .WORD .S.
015556 041227 MOVE B 0, MEM ;
015560 000017 MOVE MEM,OUT1 <CSR7> ;REG. ADDRESS.
015560 000420 MICPC=MICPC+1
015562 000018 .WORD .S.
015562 104400 SBR ERROR ;REPORT DATA ERROR.
015564 000019 MICPC=MICPC+1
015564 060604 MOVE SPAD <4>,BREG ;RESTORE BREG...
015566 000020 .WORD .S.
015566 100403 SBR 25 ;LOOP ON ERROR...
015570 000021 MICPC=MICPC+1
015570 000424 .WORD .S.
015572 000022 CALL SCP1 ;IS LOOP DATA SET.
015572 104427 MOVE B <MICPC+3>,BREG
015572 104427 MICPC=MICPC+1
015572 104427 .WORD .S.

```

35:

3588	015574	000024	MOVE SPAD <4>, BREG	
3589		060604	MICPC=MICPC+1	
3590	015574	060604	.WORD .S.	
3591	015576	000025	SBR 25	; YES, DO IT.
3592		100403	MICPC=MICPC+1	
3593	015576	100403	.WORD .S.	
3594	015600	000026	MOVE SPAD <4>, BREG	
3595		060604	MICPC=MICPC+1	
3596	015600	060604	.WORD .S.	
3597	015602	000027	SHFBIT	; NO, CONTINUE...
3598		061620	MICPC=MICPC+1	
3599	015602	061620	.WORD SBR!..SELB!..DBRSH	
3600	015604	000028	BZ 45	; IS IT DONE?...
3601		103432	MICPC=MICPC+1	
3602	015604	103432	.WORD .S.	
3603	015606	000029	SBR 25	; NO, CONTINUE...
3604		100403	MICPC=MICPC+1	
3605	015606	100403	.WORD .S.	
3606	015610			
3607	015610			
3608	015610			
3609		000032	45: SFL0T OUT0, INP0, 0, 0, 0, 115, 125, 135, 145	
3610	015610	000577	115: MOVE # 177, BREG ; START WITH BIT 7.	
3611	015612		MICPC=MICPC+1	
3612	015612		.WORD .S.	
3613		000033	1: MOVE BREG, OUT0 <0> ; SET THE BIT.	
3614	015612	062220	MICPC=MICPC+1	
3615	015614		.WORD .S.	
3616		000034	MOVE INP0 <0>, SPAD <0> ; GET THE "FOUND" IN SCRATCH PAD.	
3617	015614	023000	MICPC=MICPC+1	
3618	015616		.WORD .S.	
3619		000035	MOVE BREG, SPAD <4>	
3620	015616	063224	MICPC=MICPC+1	
3621	015620		.WORD .S.	
3622			SIFEB BREG, SPAD <0> 135 ; CHECK THE DATA...	
3623	015620			
3624		000036	SUB2C SPAD <0>, BREG, NOP	
3625	015620	060360	MICPC=MICPC+1	
3626	015622		.WORD .S.	
3627		000037	BZ 135	
3628	015622	101452	MICPC=MICPC+1	
3629	015624		.WORD .S.	
3630		000040	MOVE BREG, OUT1 <CSR4> ; GOOD DATA...	
3631	015624	061224	MICPC=MICPC+1	
3632	015626		.WORD .S.	
3633		000041	MOVE INP0 <0>, OUT1 <CSR5>; BAD DATA...	
3634	015626	021005	MICPC=MICPC+1	
3635	015630		.WORD .S.	
3636		000042	MOVE # 1, MEM ; TYPE OF ERROR...	
3637	015630	002401	MICPC=MICPC+1	
3638	015632		.WORD .S.	
3639		000043	MOVE MEM, OUT1 <CSR3> ;	
3640	015632	041223	MICPC=MICPC+1	
3641	015634		.WORD .S.	
3642		000044	MOVE # 0, MEM ;	
3643			MICPC=MICPC+1	

3644	015634	002400	.WORD .S.	
3645	015636		MOVE MEM,OUT1 (CSR7)	;REG. ADDRESS.
3646		000045	MICPC=MICPC+1	
3647	015636	041227	.WORD .S.	
3648	015640		CALL EROR	;REPORT DATA ERROR.
3649	015640		MOVE # (MICPC+3),BREG	
3650		000046	MICPC=MICPC+1	
3651	015640	000450	.WORD .S.	
3652	015642		SBR EROR	
3653		000047	MICPC=MICPC+1	
3654	015642	104400	.WORD .S.	
3655	015644		MOVE SPAD <4>,BREG	;RESTORE BREG...
3656		000050	MICPC=MICPC+1	
3657	015644	060604	.WORD .S.	
3658	015646		SBR 12\$;LOOP ON ERROR...
3659		000051	MICPC=MICPC+1	
3660	015646	100433	.WORD .S.	
3661	015650		CALL SCP1	;IS LOOP DATA SET.
3662	015650		MOVE # (MICPC+3),BREG	
3663		000052	MICPC=MICPC+1	
3664	015650	000454	.WORD .S.	
3665	015652		SBR SCP1	
3666		000053	MICPC=MICPC+1	
3667	015652	104427	.WORD .S.	
3668	015654		MOVE SPAD <4>,BREG	
3669		000054	MICPC=MICPC+1	
3670	015654	060604	.WORD .S.	
3671	015656		SBR 12\$;YES, DO IT.
3672		000055	MICPC=MICPC+1	
3673	015656	100433	.WORD .S.	
3674	015660		MOVE SPAD <4>,BREG	
3675		000056	MICPC=MICPC+1	
3676	015660	060604	.WORD .S.	
3677	015662		SHFBRT	;NO, CONTINUE...
3678		000057	MICPC=MICPC+1	
3679	015662	061620	.WORD SBR?.SELB!.DBRSH	
3680	015664		BB7 12\$;	
3681		000060	MICPC=MICPC+1	
3682	015664	103433	.WORD .S.	
3683	015666			
3684	015666		CALL SCPE	
3685	015666		MOVE # (MICPC+3),BREG	
3686		000061	MICPC=MICPC+1	
3687	015666	000463	.WORD .S.	
3688	015670		SBR SCPE	
3689		000062	MICPC=MICPC+1	
3690	015670	104454	.WORD .S.	
3691	015672		SBR 21\$	
3692		000063	MICPC=MICPC+1	
3693	015672	100400	.WORD .S.	
3694	015674		SFLT1	
3695	015674		SXZ	
3696				
3697				
3698				
3699				

;***** TEST 10 *****
;* MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.

KMC11 MICRO PROCESSOR IBUS TESTS

```

3700 ;* FLOAT A 1 THROUGH REGISTER OUT0 <1>
3701 ;* FLOAT A 0 THROUGH REGISTER OUT0 <1>
3702 015674 SXZ ;:*****
3703
3704
3705 015674 STSTN ; TEST 10
3706
3707
3708 015674 012737 000010 001202 TST10: MOV #10,STSTNM ; LOAD THE NO. OF THIS TEST
3709 015702 012737 016100 001442 MOV #TST11,NEXT ; POINT TO THE START OF NEXT TEST.
3710 ;R1 CONTAINS BASE KMC11 ADDRESS
3711 015710 004737 035536 JSR PC,LDRMT ;LOAD-VERIFY-WAIT.
3712 015714 015730 MCT10
3713 015716 104022 ERROR 22 ;TIME OUT ERROR...
3714 015720 012706 001200 MOV #STACK,SP ;RESET STACK.
3715 015724 000177 163512 JMP @NEXT ;GO TO NEXT TEST...
3716 015730 MCT10:
3717 015730 21$:
3718 015730
3719 000000 MOVE #0,BREG ;SET TO CLEAR SPAD 16
3720 015730 000400 MICPC=MICPC+1
3721 015732 .WORD .S.
3722 000001 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES...
3723 015732 063236 MICPC=MICPC+1
3724 015734 .WORD .S.
3725 015734 SFLOT OUT0,INPO,1,0,1,15,25,35,45
3726 000002 1$: MOVE #200,BREG ;START WITH BIT 7.
3727 015734 000600 MICPC=MICPC+1
3728 015736 .WORD .S.
3729 015736 2$:
3730 000003 MOVE BREG,OUT0 <1> ;SET THE BIT.
3731 015736 062221 MICPC=MICPC+1
3732 015740 .WORD .S.
3733 000004 MOVE INPO <1>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
3734 015740 023020 MICPC=MICPC+1
3735 015742 .WORD .S.
3736 000005 MOVE BREG,SPAD <4>
3737 015742 063224 MICPC=MICPC+1
3738 015744 .WORD .S.
3739 SIFEQ BREG,SPAD <0> 3$ ;CHECK THE DATA...
3740
3741 015744 SUB2C SPAD <0>,BREG,NOP
3742 000006 MICPC=MICPC+1
3743 015744 060360 .WORD .S.
3744 015746 BZ 3$
3745 000007 MICPC=MICPC+1
3746 015746 101422 .WORD .S.
3747 015750 MOVE BREG,OUT1 <CSR4> ;GOOD DATA...
3748 000010 MICPC=MICPC+1
3749 015750 061224 .WORD .S.
3750 015752 MOVE INPO <1>,OUT1 <CSR5>;BAD DATA...
3751 000011 MICPC=MICPC+1
3752 015752 021025 .WORD .S.
3753 015754 MOVE #1,MEM ;TYPE OF ERROR...
3754 000012 MICPC=MICPC+1
3755 015754 002401 .WORD .S.

```

3756	015756	000013	MOVE MEM OUT1 <CSR3>	;
3757	015756	041223	MICPC=MICPC+1	
3758	015756	041223	.WORD .S.	
3759	015760	000014	MOVE #1 MEM	;
3760	015760	002401	MICPC=MICPC+1	
3761	015760	002401	.WORD .S.	
3762	015762	000015	MOVE MEM OUT1 <CSR7>	;REG. ADDRESS.
3763	015762	041227	MICPC=MICPC+1	
3764	015762	041227	.WORD .S.	
3765	015764		CALL EROR	;REPORT DATA ERROR.
3766	015764		MOVE # <MICPC+3>, BREG	
3767	015764	000016	MICPC=MICPC+1	
3768	015764	000420	.WORD .S.	
3769	015766		SBR EROR	
3770	015766	000017	MICPC=MICPC+1	
3771	015766	104400	.WORD .S.	
3772	015770		MOVE SPAD <4>, BREG	;RESTORE BREG...
3773	015770	000020	MICPC=MICPC+1	
3774	015770	060604	.WORD .S.	
3775	015772		SBR 25	;LOOP ON ERROR...
3776	015772	000021	MICPC=MICPC+1	
3777	015772	100403	.WORD .S.	
3778	015774		CAI SCP1	;IS LOOP DATA SET.
3779	015774		MOVE # <MICPC+3>, BREG	
3780	015774	000022	MICPC=MICPC+1	
3781	015774	000424	.WORD .S.	
3782	015776		SBR SCP1	
3783	015776	000023	MICPC=MICPC+1	
3784	015776	104427	.WORD .S.	
3785	016000		MOVE SPAD <4>, BREG	
3786	016000	000024	MICPC=MICPC+1	
3787	016000	060604	.WORD .S.	
3788	016002		SBR 25	;YES, DO IT.
3789	016002	000025	MICPC=MICPC+1	
3790	016002	100403	.WORD .S.	
3791	016004		MOVE SPAD <4>, BREG	
3792	016004	000026	MICPC=MICPC+1	
3793	016004	060604	.WORD .S.	
3794	016006		SFBR	;NO, CONTINUE...
3795	016006	000027	MICPC=MICPC+1	
3796	016006	061620	.WORD SBR! .SELB! .DBRSH	
3797	016010		BB7 45	;IS IT DONE?...
3798	016010	000030	MICPC=MICPC+1	
3799	016010	103432	.WORD .S.	
3800	016012		SBR 25	;NO, CONTINUE...
3801	016012	000031	MICPC=MICPC+1	
3802	016012	100403	.WORD .S.	
3803	016014			
3804	016014			
3805	016014			
3806	016014	000032		
3807	016014	000577		
3808	016016			
3809	016016			
3810	016016	000033		
3811	016016	062221		

45: SFLOT OUTO, INPO, 1, 0, 0, 115, 125, 135, 145
 115: MOVE #177, BREG ;START WITH BIT 7.
 MICPC=MICPC+1
 .WORD .S.
 125: MOVE BREG, OUTO <1> ;SET THE BIT.
 MICPC=MICPC+1
 .WORD .S.

3812	016020		MOVE INPO <1>,SPAD <0>	;GET THE "FOUND" IN SCRATCH PAD.
3813		000034	MICPC=MICPC+1	
3814	016020	023020	.WORD .S.	
3815	016022		MOVE BREG,SPAD <4>	
3816		000035	MICPC=MICPC+1	
3817	016022	063224	.WORD .S.	
3818	016024		SIFEQ BREG,SPAD <0> 135	;CHECK THE DATA...
3819				
3820				
3821	016024		SUB2C SPAD <0>,BREG,NOP	
3822		000036	MICPC=MICPC+1	
3823	016024	060360	.WORD .S.	
3824	016026		BZ 135	
3825		000037	MICPC=MICPC+1	
3826	016026	101452	.WORD .S.	
3827	016030		MOVE BREG,OUT1 <CSR4>	;GOOD DATA...
3828		000040	MICPC=MICPC+1	
3829	016030	061224	.WORD .S.	
3830	016032		MOVE INPO <1>,OUT1 <CSR5>;BAD DATA...	
3831		000041	MICPC=MICPC+1	
3832	016032	021025	.WORD .S.	
3833	016034		MOVE #1,MEM	;TYPE OF ERROR...
3834		000042	MICPC=MICPC+1	
3835	016034	002401	.WORD .S.	
3836	016036		MOVE MEM,OUT1 <CSR3>	;
3837		000043	MICPC=MICPC+1	
3838	016036	041223	.WORD .S.	
3839	016040		MOVE #1,MEM	;
3840		000044	MICPC=MICPC+1	
3841	016040	002401	.WORD .S.	
3842	016042		MOVE MEM,OUT1 <CSR7>	;REG. ADDRESS.
3843		000045	MICPC=MICPC+1	
3844	016042	041227	.WORD .S.	
3845	016044		CALL ERROR	;REPORT DATA ERROR.
3846	016044		MOVE # <MICPC+3>,BREG	
3847		000046	MICPC=MICPC+1	
3848	016044	000450	.WORD .S.	
3849	016046		SBR ERROR	
3850		000047	MICPC=MICPC+1	
	016046	104400	.WORD .S.	
	016050		MOVE SPAD <4>,BREG	;RESTORE BREG...
3854	016050	000050	MICPC=MICPC+1	
3855	016052	060604	.WORD .S.	
3856		000051	SBR 125	;LOOP ON ERROR...
3857	016052	100433	MICPC=MICPC+1	
3858	016054		.WORD .S.	
3859	016054		CALL SCP1	;IS LOOP DATA SET.
3860		000052	MOVE # <MICPC+3>,BREG	
3861	016054	000454	MICPC=MICPC+1	
3862	016056		.WORD .S.	
3863		000053	SBR SCP1	
3864	016056	104427	MICPC=MICPC+1	
3865	016060		.WORD .S.	
3866		000054	MOVE SPAD <4>,BREG	
3867	016060	060604	MICPC=MICPC+1	
			.WORD .S.	

135:

27

```

3868 016062          SBR      125          ;YES, DO IT.
3869          000055  MICPC=MICPC+1
3870 016062 100433  .WORD    .S.
3871 016064          MOVE    SPAD <4>,BREG
3872          000056  MICPC=MICPC+1
3873 016064 060604  .WORD    .S.
3874 016066          SHFBRT          ;NO, CONTINUE...
3875          000057  MICPC=MICPC+1
3876 016066 061620  .WORD    SBR!..SELB!..DBRSH
3877 016070          SBR      125          ;
3878          000060  MICPC=MICPC+1
3879 016070 103433  .WORD    .S.
3880 016072          14S:
3881 016072          CALL    SCPE
3882 016072          MOVE    8 <MICPC+3>,BREG
3883          000061  MICPC=MICPC+1
3884 016072 000463  .WORD    .S.
3885 016074          SBR      SCPE
3886          000062  MICPC=MICPC+1
3887 016074 104454  .WORD    .S.
3888 016076          SBR      21S
3889          000063  MICPC=MICPC+1
3890 016076 100400  .WORD    .S.
3891 016100          SFLT1
3892 016100          SXZ
3893
3894
3895          ;***** TEST 11 *****
3896          ;* MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
3897          ;* FLOAT A 1 THROUGH REGISTER OUTO <2>
3898          ;* FLOAT A 0THROUGH REGISTER OUTO <2>
3899 016100          SXZ
3900
3901          ;*****
3902 016100          STSTN
3903          ; TEST 11
3904          -----
3905 016100 012737 000011 001202 TST11: MOV     #11,STSTNM          ; LOAD THE NO. OF THIS TEST
3906 016106 012737 016304 001442 MOV     #TST12,NEXT      ; POINT TO THE START OF NEXT TEST.
3907          ;R1 CONTAINS BASE KMC11 ADDRESS
3908 016114 004737 035536 JSR     PC,LDVRWT       ;LOAD-VERIFY-WAIT.
3909 016120 016134 MCT11
3910 016122 104022 ERROR    22             ;TIME OUT ERROR...
3911 016124 012706 001200 MOV     #STACK,SP      ;RESET STACK...
3912 016130 000177 163306 JMP     #NEXT          ;GO TO NEXT TEST...
3913 016134 MCT11:
3914 016134 21S:
3915 016134          MOVE    # 0,BREG      ;SET TO CLEAR SPAD 16
3916          000000  MICPC=MICPC+1
3917 016134 000400  .WORD    .S.
3918 016136          MOVE    BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES...
3919          000001  MICPC=MICPC+1
3920 016136 063236  .WORD    .S.
3921 016140          SFLOT  OUTO,INPO,2,0,1,1S,2S,3S,4S
3922 016140          1S: MOVE    # 200,BREG ;START WITH BIT 7.
3923          000002  MICPC=MICPC+1

```

KMC11 MICRO PROCESSOR IBUS TESTS

3924	016140	000600		.WORD .S.	
3925	016142		25:	MOVE BREG,OUT0 <2>	;SET THE BIT.
3926	016142			MICPC=MICPC+1	
3927		000003		.WORD .S.	
3928	016142	062222		MOVE INPO <2>,SPAD <0>	;GET THE "FOUND" IN SCRATCH PAD.
3929	016144			MICPC=MICPC+1	
3930		000004		.WORD .S.	
3931	016144	023040		MOVE BREG,SPAD <4>	
3932	016146			MICPC=MICPC+1	
3933		000005		.WORD .S.	
3934	016146	063224		SIFEQ BREG,SPAD <0> 35	;CHECK THE DATA...
3935	016150				
3936				SUBC SPAD <0>,BREG,NOP	
3937				MICPC=MICPC+1	
3938	016150			.WORD .S.	
3939		000006		BZ 35	
3940	016150	060360		MICPC=MICPC+1	
3941	016152			.WORD .S.	
3942		000007		MOVE BREG,OUT1 <CSR4>	;GOOD DATA...
3943	016152	101422		MICPC=MICPC+1	
3944	016154			.WORD .S.	
3945		000010		MOVE INPO <2>,OUT1 <CSR5>	;BAD DATA...
3946	016154	061224		MICPC=MICPC+1	
3947	016156			.WORD .S.	
3948		000011		MOVE #1,MEM	;TYPE OF ERROR...
3949	016156	021045		MICPC=MICPC+1	
3950	016160			.WORD .S.	
3951		000012		MOVE MEM,OUT1 <CSR3>	
3952	016160	002401		MICPC=MICPC+1	
3953	016162			.WORD .S.	
3954		000013		MOVE #2,MEM	
3955	016162	041223		MICPC=MICPC+1	
3956	016164			.WORD .S.	
3957		000014		MOVE MEM,OUT1 <CSR7>	;REG. ADDRESS.
3958	016164	002402		MICPC=MICPC+1	
3959	016166			.WORD .S.	
3960		000015		CALL EROR	;REPORT DATA ERROR.
3961	016166	041227		MOVE # <MICPC+3>,BREG	
3962	016170			MICPC=MICPC+1	
3963	016170			.WORD .S.	
3964		000016		SBR EROR	
3965	016170	000420		MICPC=MICPC+1	
3966	016172			.WORD .S.	
3967		000017		MOVE SPAD <4>,BREG	;RESTORE BREG...
3968	016172	104400		MICPC=MICPC+1	
3969	016174			.WORD .S.	
3970		000020		SBR 25	;LOOP ON ERROR...
3971	016174	060604		MICPC=MICPC+1	
3972	016176			.WORD .S.	
3973		000021		CALL SCP1	;IS LOOP DATA SET.
3974	016176	100403		MOVE # <MICPC+3>,BREG	
3975	016200		35:	MICPC=MICPC+1	
3976	016200			.WORD .S.	
3977		000022		SBR SCP1	
3978	016200	000424			
3979	016202				

KMC11 MICRO PROCESSOR IBUS TESTS

3980 000023
3981 016202 104427
3982 016204
3983 000024
3984 016204 060604
3985 016206
3986 000025
3987 016206 100403
3988 016210
3989 000026
3990 016210 060604
3991 016212
3992 000027
3993 016212 061620
3994 016214
3995 000030
3996 016214 103432
3997 016216
3998 000031
3999 016216 100403
4000 016220
4001 016220
4002 016220
4003 000032
4004 016220 000577
4005 016222
4006 016222
4007 000033
4008 016222 062222
4009 016224
4010 000034
4011 016224 023040
4012 016226
4013 000035
4014 016226 063224
4015 016230
4016
4017
4018 016230
4019 000036
4020 016230 060360
4021 016232
4022 000037
4023 016232 101452
4024 016234
4025 000040
4026 016234 061224
4027 016236
4028 000041
4029 016236 021045
4030 016240
4031 000042
4032 016240 002401
4033 016242
4034 000043
4035 016242 041223

```

MICPC=MICPC+1
WORD .S.
MOVE SPAD <4>,BREG
MICPC=MICPC+1
WORD .S.
SBR 25 ;YES, DO IT.
MICPC=MICPC+1
WORD .S.
MOVE SPAD <4>,BREG
MICPC=MICPC+1
WORD .S.
SHFBRT ;NO, CONTINUE...
MICPC=MICPC+1
WORD SBR!.SELB!.DBRSH
BZ 45 ;IS IT DONE?...
MICPC=MICPC+1
WORD .S.
SBR 25 ;NO, CONTINUE...
MICPC=MICPC+1
WORD .S.
45:
SFL0T OUT0,INPD,2,0,0,115,125,135,145
115: MOVE #177,BREG ;START WITH BIT 7.
MICPC=MICPC+1
WORD .S.
125:
MOVE BREG,OUT0 <2> ;SET THE BIT.
MICPC=MICPC+1
WORD .S.
MOVE INPD <2>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
MICPC=MICPC+1
WORD .S.
MOVE BREG,SPAD <4>
MICPC=MICPC+1
WORD .S.
SIFEQ BREG,SPAD <0> 135 ;CHECK THE DATA...
SUB2C SPAD <0>,BREG,NOP
MICPC=MICPC+1
WORD .S.
BZ 135
MICPC=MICPC+1
WORD .S.
MOVE BREG,OUT1 <CSR4> ;GOOD DATA...
MICPC=MICPC+1
WORD .S.
MOVE INPD <2>,OUT1 <CSR5>;BAD DATA...
MICPC=MICPC+1
WORD .S.
MOVE #1,MEM ;TYPE OF ERROR...
MICPC=MICPC+1
WORD .S.
MOVE MEM,OUT1 <CSR3>
MICPC=MICPC+1
WORD .S.

```

KMC11 MICRO PROCESSOR IBUS TESTS

4036	016244		MOVE # 2 MEM ;
4037		000044	MICPC=MICPC+1
4038	016244	002402	.WORD .S.
4039	016246		MOVE MEM,OUT1 (CSR7) ;REG. ADDRESS.
4040		000045	MICPC=MICPC+1
4041	016246	041227	.WORD .S.
4042	016250		CALL EROR ;REPORT DATA ERROR.
4043	016250		MOVE # (MICPC+3),BREG
4044		000046	MICPC=MICPC+1
4045	016250	000450	.WORD .S.
4046	016252		SBR EROR
4047		000047	MICPC=MICPC+1
4048	016252	104400	.WORD .S.
4049	016254		MOVE SPAD (4),BREG ;RESTORE BREG...
4050		000050	MICPC=MICPC+1
4051	016254	060604	.WORD .S.
4052	016256		SBR 12\$;LOOP ON ERROR...
4053		000051	MICPC=MICPC+1
4054	016256	100433	.WORD .S.
4055	016260		CALL SCP1 ;IS LOOP DATA SET.
4056	016260		MOVE # (MICPC+3),BREG
4057		000052	MICPC=MICPC+1
4058	016260	000454	.WORD .S.
4059	016262		SBR SCP1
4060		000053	MICPC=MICPC+1
4061	016262	104427	.WORD .S.
4062	016264		MOVE SPAD (4),BREG
4063		000054	MICPC=MICPC+1
4064	016264	060604	.WORD .S.
4065	016266		SBR 12\$;YES, DO IT.
4066		000055	MICPC=MICPC+1
4067	016266	100433	.WORD .S.
4068	016270		MOVE SPAD (4),BREG
4069		000056	MICPC=MICPC+1
4070	016270	060604	.WORD .S.
4071	016272		SHFBRT ;NO, CONTINUE...
4072		000057	MICPC=MICPC+1
4073	016272	061620	.WORD .SBR!..SELB!.DBRSH
4074	016274		BB7 12\$;
4075		000060	MICPC=MICPC+1
4076	016274	103433	.WORD .S.
4077	016276		CALL SCPE
4078	016276		MOVE # (MICPC+3),BREG
4079	016276		MICPC=MICPC+1
4080		000061	.WORD .S.
4081	016276	000463	SBR SCPE
4082	016300		MICPC=MICPC+1
4083		000062	.WORD .S.
4084	016300	104454	SBR 21\$
4085	016302		MICPC=MICPC+1
4086		000063	.WORD .S.
4087	016302	100400	OUTO,INPO,3,0
4088	016304		
4089	016304		
4090			
4091			

13\$:

14\$:

\$FLT1
\$XZ

KMC11 MICRO PROCESSOR IBUS TESTS

```

4092                                     ;***** TEST 12 *****
4093                                     ; MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
4094                                     ; FLOAT A 1 THROUGH REGISTER OUTO <3>
4095                                     ; FLOAT A 0 THROUGH REGISTER OUTO <3>
4096 016304 SXZ                                     ;*****
4097                                     ;*****
4098                                     ;*****
4099 016304 STSTN                                  ;
4100                                     ; TEST 12
4101                                     ;-----
4102 016304 012737 000012 001202 TST12: MOV #12,STSTNM          ; LOAD THE NO. OF THIS TEST
4103 016312 012737 016510 001442 MOV #TST13,NEXT          ; POINT TO THE START OF NEXT TEST.
4104                                     ; R1 CONTAINS BASE KMC11 ADDRESS
4105 016320 004737 035536 JSR PC,LDRWRT          ; LOAD-VERIFY-WAIT.
4106 016324 016340 MCT12
4107 016326 104022 ERROR 22          ; TIME OUT ERROR...
4108 016330 012706 001200 MOV #STACK,SP          ; RESET STACK.
4109 016334 000177 163102 JMP @NEXT          ; GO TO NEXT TEST...
4110 016340 MCT12:
4111 016340 21S:
4112 016340 MOVE #0,BREG          ; SET TO CLEAR SPAD 16
4113 000000 MICPC=MICPC+1
4114 016340 000100 .WORD .S.
4115 016342 MOVE BREG,SPAD <16> ; FOR RETURN ADDRESS PURPOSES...
4116 000001 MICPC=MICPC+1
4117 016342 063236 .WORD .S.
4118 016344 SFLOT OUTO,INPO 3,0,1,15,25,35,45
4119 016344 1S: MOVE #200,BREG ; START WITH BIT 7.
4120 000002 MICPC=MICPC+1
4121 016344 000600 .WORD .S.
4122 016346 2S:
4123 016346 MOVE BREG,OUTO <3> ; SET THE BIT.
4124 000003 MICPC=MICPC+1
4125 016346 062223 .WORD .S.
4126 016350 MOVE INPO <3>,SPAD <0> ; GET THE "FOUND" IN SCRATCH PAD.
4127 000004 MICPC=MICPC+1
4128 016350 023060 .WORD .S.
4129 016352 MOVE BREG,SPAD <4>
4130 000005 MICPC=MICPC+1
4131 016352 063224 .WORD .S.
4132 016354 $IFEQ BREG,SPAD <0> 3S ; CHECK THE DATA...
4133 016354
4134 016354
4135 016354 SUB2C SPAD <0>,BREG,NOP
4136 000006 MICPC=MICPC+1
4137 016354 060360 .WORD .S.
4138 016356 BZ 3S
4139 000007 MICPC=MICPC+1
4140 016356 101422 .WORD .S.
4141 016360 MOVE BREG,OUT1 <CSR4> ; GOOD DATA...
4142 000010 MICPC=MICPC+1
4143 016360 061224 .WORD .S.
4144 016362 MOVE INPO <3>,OUT1 <CSR5>; BAD DATA...
4145 000011 MICPC=MICPC+1
4146 016362 021065 .WORD .S.
4147 016364 MOVE #1,MEM          ; TYPE OF ERROR...

```

4148		000012	MICPC=MICPC+1	
4149	016364	002401	.WORD .S.	
4150	016366		MOVE MEM,OUT1 (CSR3)	;
4151		000013	MICPC=MICPC+1	
4152	016366	041223	.WORD .S.	
4153	016370		MOVE #3, MEM	;
4154		000014	MICPC=MICPC+1	
4155	016370	002403	.WORD .S.	
4156	016372		MOVE MEM,OUT1 (CSR7)	;REG. ADDRESS.
4157		000015	MICPC=MICPC+1	
4158	016372	041227	.WORD .S.	
4159	016374		CALL EROR	;REPORT DATA ERROR.
4160	016374		MOVE # (MICPC+3), BREG	
4161		000016	MICPC=MICPC+1	
4162	016374	000420	.WORD .S.	
4163	016376		SBR EROR	
4164		000017	MICPC=MICPC+1	
4165	016376	104400	.WORD .S.	
4166	016400		MOVE SPAD (4), BREG	;RESTORE BREG...
4167		000020	MICPC=MICPC+1	
4168	016400	060604	.WORD .S.	
4169	016402		SBR	;LOOP ON ERROR...
4170		000021	MICPC=MICPC+1	
4171	016402	100403	.WORD .S.	
4172	016404		CALL SCP1	;IS LOOP DATA SET.
4173	016404		MOVE # (MICPC+3), BREG	
4174		000022	MICPC=MICPC+1	
4175	016404	000424	.WORD .S.	
4176	016406		SBR SCP1	
4177		000023	MICPC=MICPC+1	
4178	016406	104427	.WORD .S.	
4179	016410		MOVE SPAD (4), BREG	
4180		000024	MICPC=MICPC+1	
4181	016410	060604	.WORD .S.	
4182	016412		SBR	;YES, DO IT.
4183		000025	MICPC=MICPC+1	
4184	016412	100403	.WORD .S.	
4185	016414		MOVE SPAD (4), BREG	
4186		000026	MICPC=MICPC+1	
4187	016414	060604	.WORD .S.	
4188	016416		SMBRT	;NO, CONTINUE...
4189		000027	MICPC=MICPC+1	
4190	016416	061620	.WORD .SBR!.SELB!.DBRSH	
4191	016420		BB7 45	;IS IT DONE?...
4192		000030	MICPC=MICPC+1	
4193	016420	103432	.WORD .S.	
4194	016422		SBR	;NO, CONTINUE...
4195		000031	MICPC=MICPC+1	
4196	016422	100403	.WORD .S.	
4197	016424			
4198	016424			
4199	016424			
4200		000032	SFLOT OUTO, INPO, 3, 0, 0, 115, 125, 135, 145	
4201	016424	000577	115: MOVE # 177, BREG ;START WITH BIT 7.	
4202	016426		MICPC=MICPC+1	
4203	016428		.WORD .S.	
			125: MOVE BREG, OUTO (3) ;SET THE BIT.	

KMC11 MICRO PROCESSOR IBUS TESTS

4204		000033	NICPC=NICPC+1	
4205	016436	062223	WORD .S	
4206	016430		MOVE INPO <3>,SPAD <0>	;GET THE "FOUND" IN SCRATCH PAD.
4207		000034	NICPC=NICPC+1	
4208	016430	023060	WORD .S	
4209	016432		MOVE BREG,SPAD <4>	
4210		000035	NICPC=NICPC+1	
4211	016432	063224	WORD .S	
4212	016434		\$IFEQ BREG,SPAD <0> 135	;CHECK THE DATA...
4213				
4214				
4215	016434		SUB2C SPAD <0>,BREG,NOP	
4216		000036	NICPC=NICPC+1	
4217	016434	060360	WORD .S	
4218	016436		BZ 135	
4219		000037	NICPC=NICPC+1	
4220	016436	101452	WORD .S	
4221	016440		MOVE BREG,OUT1 <CSR4>	;GOOD DATA...
4222		000040	NICPC=NICPC+1	
4223	016440	061224	WORD .S	
4224	016442		MOVE INPO <3>,OUT1 <CSR5>;BAD DATA...	
4225		000041	NICPC=NICPC+1	
4226	016442	021065	WORD .S	
4227	016444		MOVE #1,MEM	;TYPE OF ERROR...
4228		000042	NICPC=NICPC+1	
4229	016444	002401	WORD .S	
4230	016446		MOVE MEM,OUT1 <CSR3>	;
4231		000043	NICPC=NICPC+1	
4232	016446	041223	WORD .S	
4233	016450		MOVE #3,MEM	;
4234		000044	NICPC=NICPC+1	
4235	016450	002403	WORD .S	
4236	016452		MOVE MEM,OUT1 <CSR7>	;REG. ADDRESS.
4237		000045	NICPC=NICPC+1	
4238	016452	041227	WORD .S	
4239	016454		CALL EROR	;REPORT DATA ERROR.
4240	016454		MOVE # <NICPC+3>,BREG	
4241		000046	NICPC=NICPC+1	
4242	016454	000450	WORD .S	
4243	016456		SBR EROR	
4244		000047	NICPC=NICPC+1	
4245	016456	104400	WORD .S	
4246	016460		MOVE SPAD <4>,BREG	;RESTORE BREG...
4247		000050	NICPC=NICPC+1	
4248	016460	060604	WORD .S	
4249	016462		SBR 125	;LOOP ON ERROR...
4250		000051	NICPC=NICPC+1	
4251	016462	100433	WORD .S	
4252	016464		CALL SCP1	;IS LOOP DATA SET.
4253	016464		MOVE # <NICPC+3>,BREG	
4254		000052	NICPC=NICPC+1	
4255	016464	000454	WORD .S	
4256	016466		SBR SCP1	
4257		000053	NICPC=NICPC+1	
4258	016466	104427	WORD .S	
4259	016470		MOVE SPAD <4>,BREG	

135:

```

4260 000054 MICPC=MICPC+1
4261 016470 060604 .WORD .S.
4262 016472 SBR 125 ;YES, DO IT.
4263 000055 MICPC=MICPC+1
4264 016472 100433 .WORD .S.
4265 016474 MOVE SPAD <4>,BREG
4266 000056 MICPC=MICPC+1
4267 016474 060604 .WORD .S.
4268 016476 SHFRT ;NO, CONTINUE...
4269 000057 MICPC=MICPC+1
4270 016476 061620 .WORD SBR!..SELB!..DBRSH
4271 016500 BB7 125 ;
4272 000060 MICPC=MICPC+1
4273 016500 103433 .WORD .S.
4274 145: CALL SCPE
4275 016502 MOVE 0 <MICPC+3>,BREG
4276 016502 MICPC=MICPC+1
4277 000061 .WORD .S.
4278 016502 000463 SBR SCPE
4279 016504 MICPC=MICPC+1
4280 000062 .WORD .S.
4281 016504 104454 SBR 215
4282 000063 MICPC=MICPC+1
4283 016506 .WORD .S.
4284 016510 SFLT1 OUTO,INPO,4,0
4285 016510 SXZ
;***** TEST 13 *****
;# MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
;# FLOAT A 1 THROUGH REGISTER OUTO <4>
;# FLOAT A 0 THROUGH REGISTER OUTO <4>
016510 SXZ ;*****
016510 STSTN ; TEST 13
;-----
4300 016510 012737 000013 001202 TST13: MOV 013,STSTNM ; LOAD THE NO. OF THIS TEST
4301 016516 012737 016714 001442 MOV 01ST14,NEXT ; POINT TO THE START OF NEXT TEST.
4302 016524 004737 035536 JSR PC,LDRMT ;R1 CONTAINS BASE KMC11 ADDRESS
4303 016530 016544 MCT13 ;LOAD-VERIFY-WAIT.
4304 016532 104022 ERROR 22 ;TIME OUT ERROR...
4305 016534 012706 001200 MOV 01STACK,SP ;RESET STACK...
4306 016540 000177 162676 JMP 01NEXT ;GO TO NEXT TEST...
4307 016544 MCT13:
4308 016544 215:
4309 016544 MOVE 0,0,BREG ;SET TO CLEAR SPAD 16
4310 000000 MICPC=MICPC+1
4311 016544 000400 .WORD .S.
4312 016546 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES...
4313 000001 MICPC=MICPC+1
4314 016546 063236 .WORD .S.
4315 016550 SFLOT OUTO,INPO,4,0,1,15,25,35,45

```

4316	016550		15:	MOVE # 200, BREG ; START WITH BIT 7.
4317		000002		MICPC=MICPC+1
4318	016550	000600		.WORD .S.
4319	016552		25:	
4320	016552			MOVE BREG, OUTD <4> ; SET THE BIT.
4321		000003		MICPC=MICPC+1
4322	016552	062224		.WORD .S.
4323	016554			MOVE INPD <4>, SPAD <0> ; GET THE "FOUND" IN SCRATCH PAD.
4324		000004		MICPC=MICPC+1
4325	016554	023100		.WORD .S.
4326	016556			MOVE BREG, SPAD <4>
4327		000005		MICPC=MICPC+1
4328	016556	063224		.WORD .S.
4329	016560			SIFEQ BREG, SPAD <0> 35 ; CHECK THE DATA...
4330				
4331				
4332	016560			SUBC SPAD <0>, BREG, NOP
4333		000006		MICPC=MICPC+1
4334	016560	060360		.WORD .S.
4335	016562			BZ 35
4336		000007		MICPC=MICPC+1
4337	016562	101422		.WORD .S.
4338	016564			MOVE BREG, OUT1 <CSR4> ; GOOD DATA...
4339		000010		MICPC=MICPC+1
4340	016564	061224		.WORD .S.
4341	016566			MOVE INPD <4>, OUT1 <CSR5>; BAD DATA...
4342		000011		MICPC=MICPC+1
4343	016566	021105		.WORD .S.
4344	016570			MOVE # 1, MEM ; TYPE OF ERROR...
4345		000012		MICPC=MICPC+1
4346	016570	002401		.WORD .S.
4347	016572			MOVE MEM, OUT1 <CSR3> ;
4348		000013		MICPC=MICPC+1
4349	016572	041223		.WORD .S.
4350	016574			MOVE # 4, MEM ;
4351		000014		MICPC=MICPC+1
4352	016574	002404		.WORD .S.
4353	016576			MOVE MEM, OUT1 <CSR7> ; REG. ADDRESS.
4354		000015		MICPC=MICPC+1
4355	016576	041227		.WORD .S.
4356	016600			CALL EROR ; REPORT DATA ERROR.
4357	016600			MOVE # <MICPC+3>, BREG
4358		000016		MICPC=MICPC+1
4359	016600	000420		.WORD .S.
4360	016602			SBR EROR
4361		000017		MICPC=MICPC+1
4362	016602	104400		.WORD .S.
4363	016604			MOVE SPAD <4>, BREG ; RESTORE BREG...
4364		000020		MICPC=MICPC+1
4365	016604	060604		.WORD .S.
4366	016606			SBR 25 ; LOOP ON ERROR...
4367		000021		MICPC=MICPC+1
4368	016606	100403		.WORD .S.
4369	016610		35:	CALL SCP1 ; IS LOOP DATA SET.
4370	016610			MOVE # <MICPC+3>, BREG
4371		000022		MICPC=MICPC+1

KMC11 MICRO PROCESSOR IBUS TESTS

4372	016610	000424	.WORD .S.	
4373	016612		SBR SCP1	
4374		000023	NICPC=NICPC+1	
4375	016612	104427	.WORD .S.	
4376	016614		MOVE SPAD <4>,BREG	
4377		000024	NICPC=NICPC+1	
4378	016614	060604	.WORD .S.	
4379	016616		SBR 25	;YES, DO IT.
4380		000025	NICPC=NICPC+1	
4381	016616	100403	.WORD .S.	
4382	016620		MOVE SPAD <4>,BREG	
4383		000026	NICPC=NICPC+1	
4384	016620	060604	.WORD .S.	
4385	016622		SHFBRT	;NO, CONTINUE...
4386		000027	NICPC=NICPC+1	
4387	016622	061620	.WORD SBR!.SELB!.DBRSH	
4388	016624		BZ 45	;IS IT DONE?...
4389		000030	NICPC=NICPC+1	
4390	016624	103432	.WORD .S.	
4391	016626		SBR 25	;NO, CONTINUE...
4392		000031	NICPC=NICPC+1	
4393	016626	100403	.WORD .S.	
4394	016630			
4395	016630			
4396	016630			
4397		000032		
4398	016630	000577		
4399	016632			
4400	016632			
4401		000033		
4402	016632	062224		
4403	016634			
4404		000034		
4405	016634	023100		
4406	016636			
4407		000035		
4408	016636	063224		
4409	016640			
4410				
4411				
4412	016640			
4413		000036		
4414	016640	060360		
4415	016642			
4416		000037		
4417	016642	101452		
4418	016644			
4419		000040		
4420	016644	061224		
4421	016646			
4422		000041		
4423	016646	021105		
4424	016650			
4425		000042		
4426	016650	002401		
4427	016652			

```

45: SFL0T OUT0,INPO,4,0,0,115,125,135,145
115: MOVE #177,BREG ;START WITH BIT 7.
      NICPC=NICPC+1
      .WORD .S.
125: MOVE BREG,OUT0 <4> ;SET THE BIT.
      NICPC=NICPC+1
      .WORD .S.
      MOVE INPO <4>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
      NICPC=NICPC+1
      .WORD .S.
      MOVE BREG,SPAD <4>
      NICPC=NICPC+1
      .WORD .S.
      SIFEQ BREG,SPAD <0> 135 ;CHECK THE DATA...

SUB2C SPAD <0>,BREG,NOP
NICPC=NICPC+1
.WORD .S.
BZ 135
NICPC=NICPC+1
.WORD .S.
MOVE BREG,OUT1 <CSR4> ;GOOD DATA...
NICPC=NICPC+1
.WORD .S.
MOVE INPO <4>,OUT1 <CSR5>;BAD DATA...
NICPC=NICPC+1
.WORD .S.
MOVE #1,MEM ;TYPE OF ERROR...
NICPC=NICPC+1
.WORD .S.
VE MEM,OUT1 <CSR3> ;

```

KMC11 MICRO PROCESSOR IBUS TESTS

4429		000043	MICPC=MICPC+1	
4430	016652	041223	.WORD .S.	
4431	016654	000044	MOVE #4, MEM ;	
4432	016654	002404	MICPC=MICPC+1	
4433	016656		.WORD .S.	
4434		000045	MOVE MEM, OUT1 (CSR7)	;REG. ADDRESS.
4435	016656	041227	MICPC=MICPC+1	
4436	016660		.WORD .S.	
4437	016660		CALL ERROR	;REPORT DATA ERROR.
4438		000046	MOVE # (MICPC+3), BREG	
4439	016660	000450	MICPC=MICPC+1	
4440	016662		.WORD .S.	
4441		000047	SBR ERROR	
4442	016662	104400	MICPC=MICPC+1	
4443	016664		.WORD .S.	
4444		000050	MOVE SPAD (4), BREG	;RESTORE BREG...
4445	016664	060604	MICPC=MICPC+1	
4446	016666		.WORD .S.	
4447		000051	SBR 12\$;LOOP ON ERROR...
4448	016666	100433	MICPC=MICPC+1	
4449	016670		.WORD .S.	
4450	016670		CALL SCP1	;IS LOOP DATA SET.
4451		000052	MOVE # (MICPC+3), BREG	
4452	016670	000454	MICPC=MICPC+1	
4453	016672		.WORD .S.	
4454		000053	SBR SCP1	
4455	016672	104427	MICPC=MICPC+1	
4456	016674		.WORD .S.	
4457		000054	MOVE SPAD (4), BREG	
4458	016674	060604	MICPC=MICPC+1	
4459	016676		.WORD .S.	
4460		000055	SBR 12\$;YES, DO IT.
4461	016676	100433	MICPC=MICPC+1	
4462	016700		.WORD .S.	
4463		000056	MOVE SPAD (4), BREG	
4464	016700	060604	MICPC=MICPC+1	
4465	016702		.WORD .S.	
4466		000057	SMBRT	;NO, CONTINUE...
4467	016702	061620	MICPC=MICPC+1	
4468	016704		.WORD .SBR!.SELB!.DBRSH	
4469		000060	B87 12\$;	
4470	016704	103433	MICPC=MICPC+1	
4471	016706		.WORD .S.	
4472	016706			
4473	016706		CALL SCPE	
4474		000061	MOVE # (MICPC+3), BREG	
4475	016706	000463	MICPC=MICPC+1	
4476	016710		.WORD .S.	
4477		000062	SBR SCPE	
4478	016710	104454	MICPC=MICPC+1	
4479	016712		.WORD .S.	
4480		000063	SBR 21\$	
4481	016712	100400	MICPC=MICPC+1	
4482	016714		.WORD .S.	
4483	016714		SFLT1 OUT0, INP0, S, 0	
			SXZ	

4484
4485
4486
4487
4488
4489
4490 016714 SXZ
4491
4492
4493 016714 STSTN
4494
4495
4496 016714 012737 000014 001202 TST14:
4497 016722 012737 017120 001442
4498
4499 016730 004737 035536 JSR PC,LDVRWT
4500 016734 016750 MCT14
4501 016736 104022 ERROR 22
4502 016740 012706 001200 MOV #STACK,SP
4503 016744 000177 162472 JMP @NEXT
4504 016750 MCT14:
4505 016750 21S:
4506 016750
4507 000000
4508 016750 000400
4509 016752
4510 000001
4511 016752 063236
4512 016754
4513 016754
4514 000002
4515 016754 000600
4516 016756
4517 016756
4518 000003
4519 016756 062225
4520 016760
4521 000004
4522 016760 023120
4523 016762
4524 000005
4525 016762 063224
4526 016764
4527
4528
4529 016764
4530 000006
4531 016764 060360
4532 016766
4533 000007
4534 016766 101422
4535 016770
4536 000010
4537 016770 061224
4538 016772
4539 000011

```

***** TEST 14 *****
* MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
* FLOAT A 1 THROUGH REGISTER OUTO <5>
* FLOAT A 0THROUGH REGISTER OUTO <5>
:*****

```

```

TEST 14
-----
MOV #14,STSTNM ; LOAD THE NO. OF THIS TEST
MOV #TST15,NEXT ; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
;LOAD-VERIFY-WAIT.
JSR PC,LDVRWT
MCT14
ERROR 22 ;TIME OUT ERROR...
MOV #STACK,SP ;RESET STACK.
JMP @NEXT ;GO TO NEXT TEST...

MOVE #0,BREG ;SET TO CLEAR SPAD 16
MICPC=MICPC+1
.WORD .S.
MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES...
MICPC=MICPC+1
.WORD .S.
SFLOT OUTO,INPO 5,0,1,15,25,35,45
1S: MOVE #200,BREG ;START WITH BIT 7.
MICPC=MICPC+1
.WORD .S.

2S: MOVE BREG,OUTO <5> ;SET THE BIT.
MICPC=MICPC+1
.WORD .S.
MOVE INPO <5>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
MICPC=MICPC+1
.WORD .S.
MOVE BREG,SPAD <4>
MICPC=MICPC+1
.WORD .S.
SIFE0 BREG,SPAD <0> 3S ;CHECK THE DATA...

SUB2C SPAD <0>,BREG,NOP
MICPC=MICPC+1
.WORD .S.
BZ 3S
MICPC=MICPC+1
.WORD .S.
MOVE BREG,OUT1 <CSR4> ;GOOD DATA...
MICPC=MICPC+1
.WORD .S.
MOVE INPO <5>,OUT1 <CSR5>;BAD DATA...
MICPC=MICPC+1

```

4540	016772	021125	.WORD .S.	
4541	016774		MOVE # 1, MEM	;TYPE OF ERROR...
4542		000012	MICPC=MICPC+1	
4543	016774	002401	.WORD .S.	
4544	016776		MOVE MEM OUT1 <CSR3>	;
4545		000013	MICPC=MICPC+1	
4546	016776	041223	.WORD .S.	
4547	017000		MOVE # 5, MEM	;
4548		000014	MICPC=MICPC+1	
4549	017000	002405	.WORD .S.	
4550	017002		MOVE MEM OUT1 <CSR7>	;REG. ADDRESS.
4551		000015	MICPC=MICPC+1	
4552	017002	041227	.WORD .S.	
4553	017004		CALL ERROR	;REPORT DATA ERROR.
4554	017004		MOVE # <MICPC+3>, BREG	
4555		000016	MICPC=MICPC+1	
4556	017004	000420	.WORD .S.	
4557	017006		SBR ERROR	
4558		000017	MICPC=MICPC+1	
4559	017006	104400	.WORD .S.	
4560	017010		MOVE SPAD <4>, BREG	;RESTORE BREG...
4561		000020	MICPC=MICPC+1	
4562	017010	060604	.WORD .S.	
4563	017012		SBR 25	;LOOP ON ERROR...
4564		000021	MICPC=MICPC+1	
4565	017012	100403	.WORD .S.	
4566	017014		CALL SCP1	;IS LOOP DATA SET.
4567	017014		MOVE # <MICPC+3>, BREG	
4568		000022	MICPC=MICPC+1	
4569	017014	000424	.WORD .S.	
4570	017016		SBR SCP1	
4571		000023	MICPC=MICPC+1	
4572	017016	104427	.WORD .S.	
4573	017020		MOVE SPAD <4>, BREG	
4574		000024	MICPC=MICPC+1	
4575	017020	060604	.WORD .S.	
4576	017022		SBR 25	;YES, DO IT.
4577		000025	MICPC=MICPC+1	
4578	017022	100403	.WORD .S.	
4579	017024		MOVE SPAD <4>, BREG	
4580		000026	MICPC=MICPC+1	
4581	017024	060604	.WORD .S.	
4582	017026		SMBRT	;NO, CONTINUE...
4583		000027	MICPC=MICPC+1	
4584	017026	061620	.WORD .SBR!.SELB!.DBRSH	
4585	017030		BRT 45	;IS IT DONE?...
4586		000030	MICPC=MICPC+1	
4587	017030	103432	.WORD .S.	
4588	017032		SBR 25	;NO, CONTINUE...
4589		000031	MICPC=MICPC+1	
4590	017032	100403	.WORD .S.	
4591	017034			
4592	017034			
4593	017034			
4594		000032		
4595	017034	000577		

35: SFL0T OUT0, INP0, 5, 0, 0, 115, 125, 135, 145
115: MOVE # 177, BREG ;START WITH BIT 7.
MICPC=MICPC+1
.WORD .S.

4596 017036
4597 017036
4598 000033
4599 017036 062225
4600 017040
4601 000034
4602 017040 023120
4603 017042
4604 000035
4605 017042 063224
4606 017044
4607
4608
4609 017044
4610 000036
4611 017044 060360
4612 017046
4613 000037
4614 017046 101452
4615 017050
4616 000040
4617 017050 061224
4618 017052
4619 000041
4620 017052 021125
4621 017054
4622 000042
4623 017054 002401
4624 017056
4625 000043
4626 017056 041223
4627 017060
4628 000044
4629 017060 002405
4630 017062
4631 000045
4632 017062 041227
4633 017064
4634 017064
4635 000046
4636 017064 000450
4637 017066
4638 000047
4639 017066 104400
4640 017070
4641 000050
4642 017070 060604
4643 017072
4644 000051
4645 017072 100433
4646 017074
4647 017074
4648 000052
4649 017074 000454
4650 017076
4651 000053

12S:

```

MOVE BREG,OUT0 <5> ;SET THE BIT.
MICPC=MICPC+1
.WORD .S.
MOVE INPO <5>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
MICPC=MICPC+1
.WORD .S.
MOVE BREG,SPAD <4>
MICPC=MICPC+1
.WORD .S.
SIFEQ BREG,SPAD <0> 13S ;CHECK THE DATA...

SUB2C SPAD <0>,BREG,NOP
MICPC=MICPC+1
.WORD .S.
BZ 13S
MICPC=MICPC+1
.WORD .S.
MOVE BREG,OUT1 <CSR4> ;GOOD DATA...
MICPC=MICPC+1
.WORD .S.
MOVE INPO <5>,OUT1 <CSR5>;BAD DATA...
MICPC=MICPC+1
.WORD .S.
MOVE 8 I, MEM ;TYPE OF ERROR...
MICPC=MICPC+1
.WORD .S.
MOVE MEM,OUT1 <CSR3> ;
MICPC=MICPC+1
.WORD .S.
MOVE 8 S, MEM ;
MICPC=MICPC+1
.WORD .S.
MOVE MEM,OUT1 <CSR7> ;REG. ADDRESS.
MICPC=MICPC+1
.WORD .S.
CALL ENOR ;REPORT DATA ERROR.
MOVE 8 <MICPC+3>,BREG
MICPC=MICPC+1
.WORD .S.
SBR ENOR
MICPC=MICPC+1
.WORD .S.
MOVE SPAD <4>,BREG ;RESTORE BREG...
MICPC=MICPC+1
.WORD .S.
SBR 12S ;LOOP ON ERROR...
MICPC=MICPC+1
.WORD .S.
CALL SCP1 ;IS LOOP DATA SET.
MOVE 8 <MICPC+3>,BREG
MICPC=MICPC+1
.WORD .S.
SBR SCP1
MICPC=MICPC+1

```

13S:

4652	017076	104427				.WORD	.S.		
4653	017100					MOVE	SPAD <4>,BREG		
4654		000054				MICPC=MICPC+1			
4655	017100	060604				.WORD	.S.		
4656	017102					SBR	125		;YES, DO IT.
4657		000055				MICPC=MICPC+1			
4658	017102	100433				.WORD	.S.		
4659	017104					MOVE	SPAD <4>,BREG		
4660		000056				MICPC=MICPC+1			
4661	017104	060604				.WORD	.S.		
4662	017106					SFBRT			;NO, CONTINUE...
4663		000057				MICPC=MICPC+1			
4664	017106	061620				.WORD	SBR!.SELB!.DBRSH		
4665	017110					BRT	125		
4666		000060				MICPC=MICPC+1			
4667	017110	103433				.WORD	.S.		
4668	017112			145:		CALL	SCPE		
4669	017112					MOVE	# <MICPC+3>,BREG		
4670	017112					MICPC=MICPC+1			
4671		000061				.WORD	.S.		
4672	017112	000463				SBR	SCPE		
4673	017114					MICPC=MICPC+1			
4674		000062				.WORD	.S.		
4675	017114	104454				SBR	215		
4676	017116					MICPC=MICPC+1			
4677		000063				.WORD	.S.		
4678	017116	100400				OUTO,INPO,6,0			
4679	017120			SFLT1					
4680	017120			SXZ					
4681									
4682									
4683									
4684									
4685									
4686									
4687	017120			SXZ					
4688									
4689									
4690	017120			STSTN					
4691									
4692									
4693	017120	012737	000015	001202	TST15:	MOV	#15,STSTN		; LOAD THE NO. OF THIS TEST
4694	017126	012737	017324	001442		MOV	#ST16,NEXT		; POINT TO THE START OF NEXT TEST.
4695									;R1 CONTAINS BASE KMC11 ADDRESS
4696	017134	004737	035536			JSR	PC,LDVWRT		;LOAD-VERIFY-WAIT.
4697	017140	017154				MCT15			
4698	017142	104022				ERROR	22		;TIME OUT ERROR...
4699	017144	012706	001200			MOV	#STACK,SP		;RESET STACK...
4700	017150	000177	162266			JMP	#NEXT		;GO TO NEXT TEST...
4701	017154				MCT15:				
4702	017154				215:				
4703	017154					MOVE	# 0,BREG		;SET TO CLEAR SPAD 16
4704		000000				MICPC=MICPC+1			
4705	017154	000400				.WORD	.S.		
4706	017156					MOVE	BREG,SPAD <16>		;FOR RETURN ADDRESS PURPOSES...
4707		000001				MICPC=MICPC+1			

KMC11 MICRO PROCESSOR IBUS TESTS

4708 017156 063236
4709 017160
4710 017160
4711 000002
4712 017160 000600
4713 017162
4714 017162
4715 000003
4716 017162 062226
4717 017164
4718 000004
4719 017164 023140
4720 017166
4721 000005
4722 017166 063224
4723 017170
4724
4725
4726 017170
4727 000006
4728 017170 060360
4729 017172
4730 000007
4731 017172 101422
4732 017174
4733 000010
4734 017174 061224
4735 017176
4736 000011
4737 017176 021145
4738 017200
4739 000012
4740 017200 002401
4741 017202
4742 000013
4743 017202 041223
4744 017204
4745 000014
4746 017204 002406
4747 017206
4748 000015
4749 017206 041227
4750 017210
4751 017210
4752 000016
4753 017210 000420
4754 017212
4755 000017
4756 017212 104400
4757 017214
4758 000020
4759 017214 060604
4760 017216
4761 000021
4762 017216 100403
4763 017220

```

SELOT      WORD      S
           AUTO,INFO,0,1,15,25,35,45
IS:  MOVE  B 200,BREG ;START WITH BIT 7.
           MICPC=MICPC+1
           .WORD      .S.
2S:
           MOVE  BREG,OUT0 <6> ;SET THE BIT.
           MICPC=MICPC+1
           .WORD      .S.
           MOVE  INPO <6>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
           MICPC=MICPC+1
           .WORD      .S.
           MOVE  BREG,SPAD <4>
           MICPC=MICPC+1
           .WORD      .S.
           SIFEQ BREG,SPAD <0> 3S ;CHECK THE DATA...

           SUB2C SPAD <0>,BREG,NOP
           MICPC=MICPC+1
           .WORD      .S.
           BZ      3S
           MICPC=MICPC+1
           .WORD      .S.
           MOVE  BREG,OUT1 <CSR4> ;GOOD DATA...
           MICPC=MICPC+1
           .WORD      .S.
           MOVE  INPO <6>,OUT1 <CSR5>;BAD DATA...
           MICPC=MICPC+1
           .WORD      .S.
           MOVE  B 1,MEM ;TYPE OF ERROR...
           MICPC=MICPC+1
           .WORD      .S.
           MOVE  MEM,OUT1 <CSR3> ;
           MICPC=MICPC+1
           .WORD      .S.
           MOVE  B 6,MEM ;
           MICPC=MICPC+1
           .WORD      .S.
           MOVE  MEM,OUT1 <CSR7> ;REG. ADDRESS.
           MICPC=MICPC+1
           .WORD      .S.
           CALL  EROR ;REPORT DATA ERROR.
           MOVE  B <MICPC+3>,BREG
           MICPC=MICPC+1
           .WORD      .S.
           SBR  EROR
           MICPC=MICPC+1
           .WORD      .S.
           MOVE  SPAD <4>,BREG ;RESTORE BREG...
           MICPC=MICPC+1
           .WORD      .S.
           SBR  2S ;LOOP ON ERROR...
           MICPC=MICPC+1
           .WORD      .S.
3S:  CALL  SCP1 ;IS LOOP DATA SET.

```

4764 017220
 4765 000022
 4766 017220 000424
 4767 017222
 4768 000023
 4769 017222 104427
 4770 017224
 4771 000024
 4772 017224 060604
 4773 017226
 4774 000025
 4775 017226 100403
 4776 017230
 4777 000025
 4778 017230 060604
 4779 017232
 4780 000027
 4781 017232 061620
 4782 017234
 4783 000030
 4784 017234 103432
 4785 017236
 4786 000031
 4787 017236 100403
 4788 017240
 4789 017240
 4790 017240
 4791 000032
 4792 017240 000577
 4793 017242
 4794 017242
 4795 000033
 4796 017242 062226
 4797 017244
 4798 000034
 4799 017244 023140
 4800 017246
 4801 000035
 4802 017246 063224
 4803 017250
 4804
 4805
 4806 017250
 4807 000036
 4808 017250 060360
 4809 017252
 4810 000037
 4811 017252 101452
 4812 017254
 4813 000040
 4814 017254 061224
 4815 017256
 4816 000041
 4817 017256 021145
 4818 017260
 4819 000042

```

MOVE      B <MICPC+3>,BREG
MICPC=MICPC+1
.WORD     .S.
SBR       SCP1
MICPC=MICPC+1
.WORD     .S.
MOVE      SPAD <4>,BREG
MICPC=MICPC+1
.WORD     .S.
SBR       25 ;YES, DO IT.
MICPC=MICPC+1
.WORD     .S.
MOVE      SPAD <4>,BREG
MICPC=MICPC+1
.WORD     .S.
SHFBRT    ;NO, CONTINUE...
MICPC=MICPC+1
.WORD     .SBR!.SELB!.DBRSH
BB7       45 ;IS IT DONE?...
MICPC=MICPC+1
.WORD     .S.
SBR       25 ;NO, CONTINUE...
MICPC=MICPC+1
.WORD     .S.
45:
SFL0T    OUT0,INPO,6,0,0,115,125,135,145
115: MOVE      B 177,BREG ;START WITH BIT 7.
MICPC=MICPC+1
.WORD     .S.
125:
MOVE      BREG,OUT0 <6> ;SET THE BIT.
MICPC=MICPC+1
.WORD     .S.
MOVE      INPO <6>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
MICPC=MICPC+1
.WORD     .S.
MOVE      BREG,SPAD <4>
MICPC=MICPC+1
.WORD     .S.
SIFE0    BREG,SPAD <0> 135 ;CHECK THE DATA...
SUBC     SPAD <0>,BREG,NOP
MICPC=MICPC+1
.WORD     .S.
BZ       135
MICPC=MICPC+1
.WORD     .S.
MOVE      BREG,OUT1 <CSR4> ;GOOD DATA...
MICPC=MICPC+1
.WORD     .S.
MOVE      INPO <6>,OUT1 <CSR5>;BAD DATA...
MICPC=MICPC+1
.WORD     .S.
MOVE      B 1,MEM ;TYPE OF ERROR...
MICPC=MICPC+1

```

4820 017260 002401
 4821 017262
 4822 000043
 4823 017262 041223
 4824 017264
 4825 000044
 4826 017264 002406
 4827 017266
 4828 000045
 4829 017266 041227
 4830 017270
 4831 017270
 4832 000046
 4833 017270 000450
 4834 017272
 4835 000047
 4836 017272 104400
 4837 017274
 4838 000050
 4839 017274 060604
 4840 017276
 4841 000051
 4842 017276 100433
 4843 017300
 4844 017300
 4845 000052
 4846 017300 000454
 4847 017302
 4848 000053
 4849 017302 104427
 4850 017304
 4851 000054
 4852 017304 060604
 4853 017306
 4854 000055
 4855 017306 100433
 4856 017310
 4857 000056
 4858 017310 060604
 4859 017312
 4860 000057
 4861 017312 061620
 4862 017314
 4863 000060
 4864 017314 103433
 4865 017316
 4866 017316
 4867 017316
 4868 000061
 4869 017316 000463
 4870 017320
 4871 000062
 4872 017320 104454
 4873 017322
 4874 000063
 4875 017322 100400

135:

145:

```

.WORD .S.
MOVE MEM,OUT1 <CSR3> ;
MICPC=MICPC+1
.WORD .S.
MOVE 8,6, MEM ;
MICPC=MICPC+1
.WORD .S.
MOVE MEM,OUT1 <CSR7> ;REG. ADDRESS.
MICPC=MICPC+1
.WORD .S.
CALL EROR ;REPORT DATA ERROR.
MOVE 8 <MICPC+3>,BREG
MICPC=MICPC+1
.WORD .S.
SBR EROR
MICPC=MICPC+1
.WORD .S.
MOVE SPAD <4>,BREG ;RESTORE BREG...
MICPC=MICPC+1
.WORD .S.
SBR 125 ;LOOP ON ERROR...
MICPC=MICPC+1
.WORD .S.
CALL SCP1 ;IS LOOP DATA SET.
MOVE 8 <MICPC+3>,BREG
MICPC=MICPC+1
.WORD .S.
SBR SCP1
MICPC=MICPC+1
.WORD .S.
MOVE SPAD <4>,BREG
MICPC=MICPC+1
.WORD .S.
SBR 125 ;YES, DO IT.
MICPC=MICPC+1
.WORD .S.
MOVE SPAD <4>,BREG
MICPC=MICPC+1
.WORD .S.
SHFBRT ;NO, CONTINUE...
MICPC=MICPC+1
.WORD SBR!..SELB!..DBRSH ;
SBR 125
MICPC=MICPC+1
.WORD .S.
CALL SCPE
MOVE 8 <MICPC+3>,BREG
MICPC=MICPC+1
.WORD .S.
SBR SCPE
MICPC=MICPC+1
.WORD .S.
SBR 215
MICPC=MICPC+1
.WORD .S.

```

KMC11 MICRO PROCESSOR IBUS TESTS

```

4876 017324 SFLT1 OUTO,INPO,7,0
4877 017324 SXZ
4878
4879
4880
4881 ;***** TEST 16 *****
4882 ;* MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
4883 ;* FLOAT A 1 THROUGH REGISTER OUTO <7>
4884 ;* FLOAT A 0 THROUGH REGISTER OUTO <7>
4885
4886 ;:*****
4887 017324 SXZ
4888
4889 STSTN
4890 ; TEST 16
4891 017324 012737 000016 001202 TST16: MOV #16,STSTNM ; LOAD THE NO. OF THIS TEST
4892 017332 012737 017530 001442 MOV #TST17,NEXT ; POINT TO THE START OF NEXT TEST.
4893 ;R1 CONTAINS BASE KMC11 ADDRESS
4894 017340 004737 035536 JSR PC,LDRMNT ;LOAD-VERIFY-WAIT.
4895 017344 017360 MCT16
4896 017346 104022 ERROR 22 ;TIME OUT ERROR...
4897 017350 012706 001200 MOV #STACK,SP ;RESET STACK
4898 017354 000177 162062 JMP @NEXT ;GO TO NEXT TEST...
4899
4900 MCT16:
4901 215: MOVE #0,BREG ;SET TO CLEAR SPAD 16
4902 017360 000400 MICPC=MICPC+1
4903 017362 .WORD .S.
4904 000001 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES...
4905 017362 063236 MICPC=MICPC+1
4906 017364 .WORD .S.
4907 017364 SFLOT OUTO,INPO,7,0,1,15,25,35,45
4908 000002 15: MOVE #200,BREG ;START WITH BIT 7.
4909 017364 000600 MICPC=MICPC+1
4910 017366 .WORD .S.
4911 017366 25:
4912 000003 MOVE BREG,OUTO <7> ;SET THE BIT.
4913 017366 062227 MICPC=MICPC+1
4914 017370 .WORD .S.
4915 000004 MOVE INPO <7>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
4916 017370 023160 MICPC=MICPC+1
4917 017372 .WORD .S.
4918 000005 MOVE BREG,SPAD <4>
4919 017372 063224 MICPC=MICPC+1
4920 017374 .WORD .S.
4921 SIFEQ BREG,SPAD <0> 35 ;CHECK THE DATA...
4922
4923 017374 SUB2C SPAD <0>,BREG,NOP
4924 000006 MICPC=MICPC+1
4925 017374 060360 .WORD .S.
4926 017376 BZ 35
4927 000007 MICPC=MICPC+1
4928 017376 101422 .WORD .S.
4929 017400 MOVE BREG,OUT1 <CSR4> ;GOOD DATA...
4930 000010 MICPC=MICPC+1
4931 017400 061224 .WORD .S.

```


KMC11 MICRO PROCESSOR IBUS TESTS

4932	017402	000011	MOVE INPO (<7>),OUT1 (CSRS);BAD DATA...
4933			MICPC=MICPC+1
4934	017402	021165	.WORD .S.
4935	017404		MOVE # 1 MEM ;TYPE OF ERROR...
4936		000012	MICPC=MICPC+1
4937	017404	002401	.WORD .S.
4938	017406		MOVE MEM,OUT1 (CSR3) ;
4939		000013	MICPC=MICPC+1
4940	017406	041223	.WORD .S.
4941	017410		MOVE # 7 MEM ;
4942		000014	MICPC=MICPC+1
4943	017410	002407	.WORD .S.
4944	017412		MOVE MEM,OUT1 (CSR7) ;REG. ADDRESS.
4945		000015	MICPC=MICPC+1
4946	017412	041227	.WORD .S.
4947	017414		CALL EROR ;REPORT DATA ERROR.
4948	017414		MOVE # (MICPC+3),BREG
4949		000016	MICPC=MICPC+1
4950	017414	000420	.WORD .S.
4951	017416		SBR EROR
4952		000017	MICPC=MICPC+1
4953	017416	104400	.WORD .S.
4954	017420		MOVE SPAD (4),BREG ;RESTORE BREG...
4955		000020	MICPC=MICPC+1
4956	017420	060604	.WORD .S.
4957	017422		SBR 25 ;LOOP ON ERROR...
4958		000021	MICPC=MICPC+1
4959	017422	100403	.WORD .S.
4960	017424		CALL SCP1 ;IS LOOP DATA SET.
4961	017424		MOVE # (MICPC+3),BREG
4962		000022	MICPC=MICPC+1
4963	017424	000424	.WORD .S.
4964	017426		SBR SCP1
4965		000023	MICPC=MICPC+1
4966	017426	104427	.WORD .S.
4967	017430		MOVE SPAD (4),BREG
4968		000024	MICPC=MICPC+1
4969	017430	060604	.WORD .S.
4970	017432		SBR 25 ;YES, DO IT.
4971		000025	MICPC=MICPC+1
4972	017432	100403	.WORD .S.
4973	017434		MOVE SPAD (4),BREG
4974		000026	MICPC=MICPC+1
4975	017434	060604	.WORD .S.
4976	017436		SFBRT ;NO, CONTINUE...
4977		000027	MICPC=MICPC+1
4978	017436	061620	.WORD .SBR!.SELB!.DBRSH
4979	017440		BB7 45 ;IS IT DONE?...
4980		000030	MICPC=MICPC+1
4981	017440	103432	.WORD .S.
4982	017442		SBR 25 ;NO, CONTINUE...
4983		000031	MICPC=MICPC+1
4984	017442	100403	.WORD .S.
4985	017444		
4986	017444		
4987	017444		

45:
SFL0T OUTD,INPO,7,0,0,115,125,135,145
115: MOVE # 177,BREG ;START WITH BIT 7.

4998 000032
4999 017444 000577
4990 017446
4991 017446
4992 000033
4993 017446 062227
4994 017450
4995 000034
4996 017450 023160
4997 017452
4998 000035
4999 017452 063224
5000 017454
5001
5002
5003 017454
5004 000036
5005 017454 060360
5006 017456
5007 000037
5008 017456 101452
5009 017460
5010 000040
5011 017460 061224
5012 017462
5013 017464 000041
5014 017464 021165
5015 017464 000042
5016 017466 002401
5017
5018
5019 000043
5020 017466 041223
5021 017470
5022 000044
5023 017470 002407
5024 017472
5025 000045
5026 017472 041227
5027 017474
5028 017474
5029
5030 000046
5031 017474 000450
5032 017476
5033 000047
5034 017476 104400
5035 017500
5036 000050
5037 017500 060604
5038 017502
5039 000051
5040 017502 100433
5041 017504
5042 000052
5043 017504 000454

```

MICPC=MICPC+1
.WORD .S.
125: MOVE BREG,OUT0 <7> ;SET THE BIT.
MICPC=MICPC+1
.WORD .S.
MOVE INPO <7>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
MICPC=MICPC+1
.WORD .S.
MOVE BREG,SPAD <4>
MICPC=MICPC+1
.WORD .S.
SIFEQ BREG,SPAD <0> 135 ;CHECK THE DATA...

SUBC SPAD <0>,BREG,NOP
MICPC=MICPC+1
.WORD .S.
BZ 135
MICPC=MICPC+1
.WORD .S.
MOVE BREG,OUT1 <CSR4> ;GOOD DATA...
MICPC=MICPC+1
.WORD .S.
MOVE INPO <7>,OUT1 <CSR5>;BAD DATA...
MICPC=MICPC+1
.WORD .S.
MOVE 8,1,MEH ;TYPE OF ERROR...
MICPC=MICPC+1
.WORD .S.
MOVE MEH,OUT1 <CSR3> ;
MICPC=MICPC+1
.WORD .S.
MOVE 8,7,MEH ;
MICPC=MICPC+1
.WORD .S.
MOVE MEH,OUT1 <CSR7> ;REG. ADDRESS.
MICPC=MICPC+1
.WORD .S.
CALL ENDR ;REPORT DATA ERROR.
MOVE 8 <MICPC+3>,BREG
MICPC=MICPC+1
.WORD .S.
SRR ENDR
MICPC=MICPC+1
.WORD .S.
MOVE SPAD <4>,BREG ;RESTORE BREG...
MICPC=MICPC+1
.WORD .S.
SRR 125 ;LOOP ON ERROR...
MICPC=MICPC+1
.WORD .S.
135: CALL SCP1 ;IS LOOP DATA SET.
MOVE 8 <MICPC+3>,BREG
MICPC=MICPC+1
.WORD .S.

```

KMC11 MICRO PROCESSOR IBUS TESTS

5044 017506
5045 000053
5046 017506 104427
5047 017510
5048 000054
5049 017510 060604
5050 017512
5051 000055
5052 017512 100433
5053 017514
5054 000056
5055 017514 060604
5056 017516
5057 000057
5058 017516 061620
5059 017520
5060 000060
5061 017520 103433
5062 017522
5063 000061
5064 017522 000463
5065 017524
5066 000062
5067 017524 104454
5068 017526
5069 000063
5070 017526 100400
5071 017530
5072 017530
5073 017530
5074 017530
5075
5076
5077
5078
5079
5080
5081
5082 017530
5083
5084
5085 017530
5086
5087
5088 017530 012737 000017 001202
5089 017536 012737 017720 001442
5090
5091 017544 004737 035536
5092 017550 017564
5093 017552 104022
5094 017554 012706 001200
5095 017560 000177 161656
5096 017564
5097 017564
5098 017564
5099 000000

```

SBR SCP1
MICPC=MICPC+1
.WORD .S.
MOVE SPAD (4),BREG
MICPC=MICPC+1
.WORD .S.
SBR 125 ;YES, DO IT.
MICPC=MICPC+1
.WORD .S.
MOVE SPAD (4),BREG
MICPC=MICPC+1
.WORD .S.
SHFRT ;NO, CONTINUE...
MICPC=MICPC+1
.WORD SBR!.SELB!.DBRSH
B87 125 ;
MICPC=MICPC+1
.WORD .S.
145:
CALL SCPE
MOVE # (MICPC+3),BREG
MICPC=MICPC+1
.WORD .S.
SBR SCPE
MICPC=MICPC+1
.WORD .S.
SBR 215
MICPC=MICPC+1
.WORD .S.
SPROC4
SXZ
;***** TEST 17 *****
;# MICRO PROCESSOR B REGISTER TEST
;# FLOAT A 1 THROUGH THE BREG.
;# FLOAT A 0 THROUGH THE BREG.
;#
SXZ
;:*****
STSTN
; TEST 17
TST17:
MOV #17,STSTN ; LOAD THE NO. OF THIS TEST
MOV #TST20,NEXT ; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
;LOAD-VERIFY-WAIT.
JSR PC,LDRMT
MCT17
ERROR 22 ;TIME OUT ERROR...
MOV #STACK,SP ;RESET STACK...
JMP @NEXT ;GO TO NEXT TEST...
MCT17:
215:
MOVE #0,BREG ;SET TO CLEAR SPAD 16.
MICPC=MICPC+1

```

5100	017564	000400	.WORD .S.	
5101	017566		MOVE BREG, SPAD <16>	;FOR RETURN ADDRESS PURPOSE...
5102		000001	MICPC=MICPC+1	
5103	017566	063236	.WORD .S.	
5104	017570		SBRFLT 1, 18, 25, 38, 45	
5105	017570		IS: MOVE # 001, MEM	;START WITH BIT0.
5106		000012	MICPC=MICPC+1	
5107	017570	002401	.WORD .S.	
5108	017572		MOVE MEM, SPAD <2>	;SET THE DATA IN SPAD0.
5109		000003	MICPC=MICPC+1	
5110	017572	043222	.WORD .S.	
5111	017574		25: MOVE MEM, BREG	;SET THE BIT IN BREG.
5112		000004	MICPC=MICPC+1	
5113	017574	040620	.WORD .S.	
5114	017576		MOVE BREG, SPAD <1>	;PUT IT IN SPAD <1>.
5115		000005	MICPC=MICPC+1	
5116	017576	063221	.WORD .S.	
5117	017600		SIFEQ MEM, SPAD <1> 35	;COMPARE DATA...
5118				
5119				
5120	017600		SUB2C SPAD <1>, MEM, NOP	
5121		000006	MICPC=MICPC+1	
5122	017600	040361	.WORD .S.	
5123	017602		BZ 25	
5124		000007	MICPC=MICPC+1	
5125	017602	101416	.WORD .S.	
5126	017604		MOVE MEM, OUT1 <CSR4>	;REPORT DATA ERROR.
5127		000010	MICPC=MICPC+1	
5128	017604	041224	.WORD .S.	
5129	017606		MOVE BREG, OUT1 <CSR5>	; " " " "
5130		000011	MICPC=MICPC+1	
5131	017606	061225	.WORD .S.	
5132	017610		MOVE # 3, OUT1 <CSR3>	;ERROR TYPE...
5133		000012	MICPC=MICPC+1	
5134	017610	001003	.WORD .S.	
5135	017612		CALL ERROR1	;DATA ERROR...
5136	017612		MOVE # <MICPC+3>, BREG	
5137		000013	MICPC=MICPC+1	
5138	017612	000415	.WORD .S.	
5139	017614		SBR ERROR1	
5140		000014	MICPC=MICPC+1	
5141	017614	104401	.WORD .S.	
5142	017616		SBR 25	;LOOP ON ERROR...
5143		000015	MICPC=MICPC+1	
5144	017616	100404	.WORD .S.	
5145	017620		35: CALL SCP11	;LOOP THE DATA...
5146	017620		MOVE # <MICPC+3>, BREG	
5147		000016	MICPC=MICPC+1	
5148	017620	000420	.WORD .S.	
5149	017622		SBR SCP11	
5150		000017	MICPC=MICPC+1	
5151	017622	104430	.WORD .S.	
5152	017624		SBR 25	;YES, LOOP...
5153		000020	MICPC=MICPC+1	
5154	017624	100404	.WORD .S.	
5155	017626		MOVE MEM, BREG	

5156		000021	NICPC=NICPC+1	
5157	017626	040620	.WORD .S.	
5158	017630		SROL SPAD (2)	;SET THE NEXT BIT.
5159		000022	NICPC=NICPC+1	
5160	017630	063142	.WORD .S.	
5161	017632		SRDC SPAD (2)	
5162		000023	NICPC=NICPC+1	
5163	017632	063102	.WORD .S.	
5164	017634		MOVE SPAD (2),MEM	
5165		000024	NICPC=NICPC+1	
5166	017634	062602	.WORD .S.	
5167	017636		BZ 45	;DONE...
5168		000025	NICPC=NICPC+1	
5169	017636	103427	.WORD .S.	
5170	017640		SBR 25	;NO, CONTINUE...
5171		000026	NICPC=NICPC+1	
5172	017640	100404	.WORD .S.	
5173	017642		4S: SERFLT 0,11S,12S,13S,14S	
5174	017642		11S: MOVE 8,376, MEM	;START WITH BIT0.
5175	017642		NICPC=NICPC+1	
5176		000027	.WORD .S.	
5177	017642	002776	MOVE MEM SPAD (2)	;SET THE DATA IN SPAD0.
5178	017644		NICPC=NICPC+1	
5179		000030	.WORD .S.	
5180	017644	043222	MOVE MEM BREG	;SET THE BIT IN BREG.
5181	017646		NICPC=NICPC+1	
5182		000031	.WORD .S.	
5183	017646	040620	MOVE BREG SPAD (1)	;PUT IT IN SPAD (1).
5184	017650		NICPC=NICPC+1	
5185		000032	.WORD .S.	
5186	017650	063221	SIFE0 MEM, SPAD (1) 13S	;COMPARE DATA...
5187	017652			
5188				
5189				
5190	017652		SUB2C SPAD (1), MEM, NOP	
5191		000033	NICPC=NICPC+1	
5192	017652	040361	.WORD .S.	
5193	017654		BZ 13S	
5194		000034	NICPC=NICPC+1	
5195	017654	101443	.WORD .S.	
5196	017656		MOVE MEM OUT1 (CSR4)	;REPORT DATA ERROR.
5197		000035	NICPC=NICPC+1	
5198	017656	041224	.WORD .S.	
5199	017660		MOVE BREG OUT1 (CSR5)	; " " " "
5200		000036	NICPC=NICPC+1	
5201	017660	061225	.WORD .S.	
5202	017662		MOVE 3, OUT1 (CSR3)	;ERROR TYPE...
5203		000037	NICPC=NICPC+1	
5204	017662	001003	.WORD .S.	
5205	017664		CALL ERROR1	;DATA ERROR...
5206	017664		MOVE 8 (NICPC+3), BREG	
5207		000040	NICPC=NICPC+1	
5208	017664	000442	.WORD .S.	
5209	017666		SBR ERROR1	
5210		000041	NICPC=NICPC+1	
5211	017666	104401	.WORD .S.	

KMC11 MICRO PROCESSOR IBUS TESTS

012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
038
039
040
041
042
043
044
045
046
047
048
049
050
051
052
053
054
055
056
057
058
059
060
061
062
063
064
065
066
067

017670 000042
017670 100431
017672 000043
017672 000445
017674 000044
017674 104430
017676 000045
017676 100431
017700 000046
017700 040620
017702 000047
017702 063142
017704 000050
017704 063102
017706 000051
017706 062602
017710 000052
017710 103431
017712
017712
017712
017712 000053
017712 000455
017714 000054
017714 104454
017716 000055
017716 100400
017720
017720
017720
017720
017720 012737 000020 001202
017726 012737 020134 001442

```

SBR 125 ;LOOP ON ERROR...
MICPC=MICPC+1
.WORD .S.
135: CALL SCP11 ;LOOP THE DATA...
MOVE 8 <MICPC+3>,BREG
MICPC=MICPC+1
.WORD .S.
SBR SCP11
MICPC=MICPC+1
.WORD .S.
SBR 125 ;YES, LOOP...
MICPC=MICPC+1
.WORD .S.
MOVE MEM,BREG
MICPC=MICPC+1
.WORD .S.
SROL SPAD <2> ;SET THE NEXT BIT.
MICPC=MICPC+1
.WORD .S.
SADC SPAD <2>
MICPC=MICPC+1
.WORD .S.
MOVE SPAD <2>,MEM ;
MICPC=MICPC+1
.WORD .S.
137 125 ;CONTINUE, IF NOT DONE...
MICPC=MICPC+1
.WORD .S.
145: CALL SCPE
MOVE 8 <MICPC+3>,BREG
MICPC=MICPC+1
.WORD .S.
SBR SCPE
MICPC=MICPC+1
.WORD .S.
SBR 215
MICPC=MICPC+1
.WORD .S.
SSPTS1 4
SXZ
;***** TEST 20 *****
; * SCRATCH PAD TEST FOR SP4
; * FLOAT A 1 THROUGH SCRATCH PAD 4
; * FLOAT A 0 THROUGH SCRATCH PAD 4
;*****
017720 SXZ
017720 STSTN
; TEST 20
-----
TST20: MOV #20,STSTN ; LOAD THE NO. OF THIS TEST
MOV #TST21,NEXT ; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
```

KMC11 SCRATCH PAD TESTS

5268	017774	004737	035536	JSR	PC,LDVRNT	;LOAD-VERIFY-WAIT.
5269	017740	017754		MCT20		
5270	017742	104022		ERROR	22	;TIME OUT ERROR...
5271	017744	012706	001200	MOV	8STACK,SP	;RESET STACK.
5272	017750	000177	161466	JMP	3NEXT	;GO TO NEXT TEST...
5273	017754			MCT20:		
5274	017754			218:		
5275	017754			MOVE	8 0,BREG	;SET TO CLEAR SPAD 16...
5276		000000		NICPC=NICPC+1		
5277	017754	000400		.WORD	.S.	
5278	017756			MOVE	BREG,SPAD <16>	;FOR RETURN ADDRESS PURPOSES.
5279		000001		NICPC=NICPC+1		
5280	017756	063236		.WORD	.S.	
5281	017760			SSPFLT	1 18,28,38,48 4	
5282	017760			15:	MOVE 8 200,BREG	;START WITH BIT 7.
5283		000002		NICPC=NICPC+1		
5284	017760	000600		.WORD	.S.	
5285	017762			25:	MOVE BREG,SPAD <4>	;LOAD THE SCRATCH PAD...
5286		000003		NICPC=NICPC+1		
5287	017762	063224		.WORD	.S.	
5288	017764			MOVE	SPAD <4>,MEM	;GET THE "FOUND"...
5289		000004		NICPC=NICPC+1		
5290	017764	062604		.WORD	.S.	
5291	017766			MOVE	MEM,SPAD <0>	;
5292		000005		NICPC=NICPC+1		
5293	017766	043220		.WORD	.S.	
5294	017770			MOVE	BREG,MEM	;SAVE THE CONTENTS OF BREG...
5295		000006		NICPC=NICPC+1		
5296	017770	062620		.WORD	.S.	
5297	017772			SIFEQ	BREG,SPAD <0> 3\$;IF GOOD.. CONTINUE...
5298		000007		SUB2C	SPAD <0>,BREG,NOP	
5299	017772	060360		NICPC=NICPC+1		
5300	017774			.WORD	.S.	
5301		000010		BZ	3\$	
5302	017774	101424		NICPC=NICPC+1		
5303	017776			.WORD	.S.	
5304		000011		MOVE	BREG,OUT1 <CSR4>	;ELSE, REPORT ERROR...
5305	017776	061224		NICPC=NICPC+1		
5306	020000			.WORD	.S.	
5307		000012		MOVE	SPAD <0>,BREG	;
5308	020000	060600		NICPC=NICPC+1		
5309		000013		.WORD	.S.	
5310	020002			MOVE	BREG,OUT1 <CSR5>	;BAD DATA...
5311	020002	061225		NICPC=NICPC+1		
5312		000014		.WORD	.S.	
5313	020004			MOVE	8 4,BREG	;TYPE OF ERROR...
5314		000015		NICPC=NICPC+1		
5315	020004	000404		.WORD	.S.	
5316		000016		MOVE	BREG,OUT1 <CSR3>	;
5317	020006			NICPC=NICPC+1		
5318		000017		.WORD	.S.	
5319	020010			MOVE	8 4,BREG	;LOAD ADDRESS.
5320	020010	000404		NICPC=NICPC+1		
5321		000018		.WORD	.S.	
5322	020010					
5323		000404				

5370	020012		MOVE BREG,OUT1 (CSR7)	;
5371	020012	000017	MICPC=MICPC+1	
5372	020014	061227	.WORD .S.	
5373	020014		CALL EROR	;DATA ERROR!!
5374	020014		MOVE # (MICPC+3),BREG	
5375	020014	000020	MICPC=MICPC+1	
5376	020016	000422	.WORD .S.	
5377	020016		SBR EROR	
5378	020016	000021	MICPC=MICPC+1	
5379	020020	104400	.WORD .S.	
5380	020020		MOVE MEM,BREG	;RESTORE BREG...
5381	020020	000022	MICPC=MICPC+1	
5382	020022	040620	.WORD .S.	
5383	020022		SBR 25	;LOOP ON ERROR...
5384	020022	000023	MICPC=MICPC+1	
5385	020024	100403	.WORD .S.	
5386	020024		CALL SCP1	;IS LOOP DATA SET???
5387	020024		MOVE # (MICPC+3),BREG	
5388	020024	000024	MICPC=MICPC+1	
5389	020026	000426	.WORD .S.	
5390	020026		SBR SCP1	
5391	020026	000025	MICPC=MICPC+1	
5392	020030	104427	.WORD .S.	
5393	020030		MOVE MEM,BREG	;RESTORE BREG...
5394	020030	000026	MICPC=MICPC+1	
5395	020032	040620	.WORD .S.	
5396	020032		SBR 25	;LOOP ON DATA.
5397	020032	000027	MICPC=MICPC+1	
5398	020034	100403	.WORD .S.	
5399	020034		MOVE MEM,BREG	;RESTORE BREG...
5400	020034	000030	MICPC=MICPC+1	
5401	020036	040620	.WORD .S.	
5402	020036	000031	SHFBRT	;SET THE NEXT BIT...
5403	020036	061620	MICPC=MICPC+1	
5404	020040		.WORD SBR!.SELB!.DBRSH	
5405	020040	000032	BB7 45	;BRANCH IF DONE...
5406	020042	103434	MICPC=MICPC+1	
5407	020042		.WORD .S.	
5408	020042	000033	SBR 25	;CONTINUE...
5409	020042	100403	MICPC=MICPC+1	
5410	020044		.WORD .S.	
5411	020044			
5412	020044			
5413	020044			
5414	020044	000034	45: SSPFLT 0,115,125,135,145,4	
5415	020044	000577	115: MOVE # 177,BREG ;START WITH BIT 7.	
5416	020046		MICPC=MICPC+1	
5417	020046		.WORD .S.	
5418	020046	000035	125: MOVE BREG,SPAD (4) ;LOAD THE SCRATCH PAD...	
5419	020050	063224	MICPC=MICPC+1	
5420	020050		.WORD .S.	
5421	020050	000036	MOVE SPAD (4),MEM ;GET THE "FOUND"...	
5422	020052	062604	MICPC=MICPC+1	
5423	020052		.WORD .S.	
5424	020052	000037	MOVE MEM,SPAD (0)	;
5425	020054	043220	MICPC=MICPC+1	
5426	020054		.WORD .S.	
5427	020054		MOVE BREG,MEM	;SAVE THE CONTENTS OF BREG...

020054
020056
020058
020060
020062
020064
020066
020068
020070
020072
020074
020076
020078
020080
020082
020084
020086
020088
020090
020092
020094
020096
020098
020100
020102
020104
020106
020108
020110
020112
020114
020116

000040
062620
000041
060360
000042
101456
000043
061224
000044
060600
000045
061225
000046
000404
000047
061223
000050
000404
000051
061227
000052
000454
000053
104400
000054
040620
000055
100435
000056
000460
000057
104427
000060
040620

```

MICPC=MICPC+1
WORD $
$IFEQ BREG,SPAD <0> 135 ;IF GOOD.. CONTINUE...

SUBPC SPAD <0>,BREG,NOP
MICPC=MICPC+1
WORD $
BZ 135
MICPC=MICPC+1
WORD $
MOVE BREG,OUT1 <CSR4> ;ELSE, REPORT ERROR...
MICPC=MICPC+1
WORD $
MOVE SPAD.<0>,BREG ;
MICPC=MICPC+1
WORD $
MOVE BREG,OUT1 <CSR5> ;BAD DATA...
MICPC=MICPC+1
WORD $
MOVE #4,BREG ;TYPE OF ERROR...
MICPC=MICPC+1
WORD $
MOVE BREG,OUT1 <CSR3> ;
MICPC=MICPC+1
WORD $
MOVE #4,BREG ;LOAD ADDRESS.
MICPC=MICPC+1
WORD $
MOVE BREG,OUT1 <CSR7> ;
MICPC=MICPC+1
WORD $
CALL EROR ;DATA ERROR!!
MOVE # <MICPC+3>,BREG
MICPC=MICPC+1
WORD $
SBR EROR
MICPC=MICPC+1
WORD $
MOVE MEM,BREG ;RESTORE BREG...
MICPC=MICPC+1
WORD $
SBR 125 ;LOOP ON ERROR...
MICPC=MICPC+1
WORD $
CALL SCP1 ;IS LOOP DATA SET??
MOVE # <MICPC+3>,BREG
MICPC=MICPC+1
WORD $
SBR SCP1
MICPC=MICPC+1
WORD $
MOVE MEM,BREG ;RESTORE BREG...
MICPC=MICPC+1
WORD $
SBR 125 ;LOOP ON DATA.

```

135:

```

5436 000061      MICPC=MICPC+1
5437 020116 100435  .WORD  .S.
5438 020120      MOVE  MEM,BREG      ;RESTORE BREG...
5439 000062      MICPC=MICPC+1
5440 020120 040620  .WORD  .S.
5441 020122      SHFBT                ;SET THE NEXT BIT...
5442 000063      MICPC=MICPC+1
5443 020122 061620  .WORD  SBR!.SELB!.DBRSH
5444 020124      BBT 128                ;BRANCH IF NOT DONE...
5445 000064      MICPC=MICPC+1
5446 020124 103435  .WORD  .S.
145:
5447 020126      CALL  SCPE
5448 020126      MOVE  # <MICPC+3>,BREG
5449 020126      MICPC=MICPC+1
5450 000065      .WORD  .S.
5451 020126 000467  SBR  SCPE
5452 020130      MICPC=MICPC+1
5453 000066      .WORD  .S.
5454 020130 104454  SBR  21$
5455 020132      MICPC=MICPC+1
5456 000067      .WORD  .S.
5457 020132 100400  $SPTS1 5
5458 020134      $XZ
5459 020134
5460
5461
5462
5463
5464
5465
5466 020134      $XZ
5467
5468
5469 020134      $STSTN
5470
5471
5472 020134 012737 000021 001202 TST21: MOV  #21,$STSTNM      ; LOAD THE NO. OF THIS TEST
5473 020142 012737 020350 001442      MOV  #TST22,NEXT      ; POINT TO THE START OF NEXT TEST.
5474
5475 020150 004737 035536      JSR  PC,LDVRNT      ;R1 CONTAINS BASE KMC11 ADDRESS
5476 020154 020170      MCT21                ;LOAD-VERIFY-WAIT.
5477 020156 104022      ERROR  22          ;TIME OUT ERROR...
5478 020160 012706 001200      MOV  #STACK,SP      ;RESET STACK...
5479 020164 000177 161252      JMP  @NEXT          ;GO TO NEXT TEST...
5480 020170      MCT21:
5481 020170      21$:
5482 020170      MOVE  # 0,BREG      ;SET TO CLEAR SPAD 16...
5483 000000      MICPC=MICPC+1
5484 020170 000400      .WORD  .S.
5485 020172      MOVE  BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES.
5486 000001      MICPC=MICPC+1
5487 020172 063236      .WORD  .S.
5488 020174      SSPFLT
5489 020174      1$:
5490 000002      MOVE  # 200,BREG ;START WITH BIT 7.
5491 020174 000600      MICPC=MICPC+1
                    .WORD  .S.

```

```

***** TEST 21 *****
* SCRATCH PAD TEST FOR SPS
* FLOAT A 1 THROUGH SCRATCH PAD 5
* FLOAT A 0 THROUGH SCRATCH PAD 5
*****

```

KMC11 SCRATCH PAD TESTS

5492
5493
5494
5495
5496
5497
5498
5499
5500
5501
5502
5503
5504
5505
5506
5507
5508
5509
5510
5511
5512
5513
5514
5515
5516
5517
5518
5519
5520
5521
5522
5523
5524
5525
5526
5527
5528
5529
5530
5531
5532
5533
5534
5535
5536
5537
5538
5539
5540
5541
5542
5543
5544
5545
5546
5547

020176 000003
020176 063225
020200 000004
020202 062605
020202 000005
020204 043220
020204 000006
020206 062620

020206 000007
020206 060360
020210 000010
020210 101424
020212 000011
020212 061224
020214 000012
020214 060600
020216 000013
020216 061225
020220 000014
020220 000404
020222 000015
020222 061223
020224 000016
020224 000405
020226 000017
020226 061227
020230 000020
020230 000422
020232 000021
020232 104400
020234 000022
020234 040620
020236 000023
020236 100403
020240

```
25:  MOVE BREG,SPAD <5> ;LOAD THE SCRATCH PAD...
      MICPC=MICPC+1
      .WORD $
      MOVE SPAD <5>,MEM ;GET THE "FOUND"...
      MICPC=MICPC+1
      .WORD $
      MOVE MEM,SPAD <0> ;
      MICPC=MICPC+1
      .WORD $
      MOVE BREG,MEM ;SAVE THE CONTENTS OF BREG...
      MICPC=MICPC+1
      .WORD $
      SIFEQ BREG,SPAD <0> 3$ ;IF GOOD.. CONTINUE...

      SUB2C SPAD <0>,BREG,NOP
      MICPC=MICPC+1
      .WORD $
      BZ 3$
      MICPC=MICPC+1
      .WORD $
      MOVE BREG,OUT1 <CSR4> ;ELSE, REPORT ERROR...
      MICPC=MICPC+1
      .WORD $
      MOVE SPAD <0>,BREG ;
      MICPC=MICPC+1
      .WORD $
      MOVE BREG,OUT1 <CSR5> ;BAD DATA...
      MICPC=MICPC+1
      .WORD $
      MOVE B 4,BREG ;TYPE OF ERROR...
      MICPC=MICPC+1
      .WORD $
      MOVE BREG,OUT1 <CSR3> ;
      MICPC=MICPC+1
      .WORD $
      MOVE B 5,BREG ;LOAD ADDRESS.
      MICPC=MICPC+1
      .WORD $
      MOVE BREG,OUT1 <CSR7> ;
      MICPC=MICPC+1
      .WORD $
      CALL EROR ;DATA ERROR!!
      MOVE B <MICPC+3>,BREG
      MICPC=MICPC+1
      .WORD $
      SBR EROR
      MICPC=MICPC+1
      .WORD $
      MOVE MEM,BREG ;RESTORE BREG...
      MICPC=MICPC+1
      .WORD $
      SBR 2$ ;LOOP ON ERROR...
      MICPC=MICPC+1
      .WORD $
      CALL S 31 ;IS LOOP DATA SET???
```

KMC11 SCRATCH PAD TESTS

020240	000024	MOVE	B (<NICPC+3>),BREG	
020240	000026	NICPC=NICPC+1		
020242	000025	.WORD	.S.	
020242	104427	SBR	SCP1	
020244	000026	NICPC=NICPC+1		
020244	040620	.WORD	.S.	;RESTORE BREG...
020246	000027	MOVE	MEM,BREG	
020246	100403	NICPC=NICPC+1		;LOOP ON DATA.
020250	000030	.WORD	.S.	;RESTORE BREG...
020250	040620	MOVE	MEM,BREG	
020252	000031	NICPC=NICPC+1		;SET THE NEXT BIT...
020252	061620	.WORD	.S.	
020254	000032	SIFBRT		;BRANCH IF DONE...
020254	103434	NICPC=NICPC+1		
020256	000033	.WORD	.S.	;CONTINUE...
020256	100403	NICPC=NICPC+1		
020258		.WORD	.S.	
020260		.WORD	.S.	
020260	000034	45:	SSPFLT 0,118,128,138,148,5	
020262	000035	118:	MOVE B 177,BREG ;START WITH BIT 7.	
020262	063225	NICPC=NICPC+1		
020264	000036	.WORD	.S.	
020264	062605	125:	MOVE BREG,SPAD <5> ;LOAD THE SCRATCH PAD...	
020266	000037	NICPC=NICPC+1		
020266	043220	.WORD	.S.	
020270	000040	MOVE	SPAD <5>,MEM ;GET THE "FOUND"...	
020270	062620	NICPC=NICPC+1		
020272		.WORD	.S.	
020272		MOVE	MEM,SPAD <0> ;	
020272	000041	NICPC=NICPC+1		
020272	060360	.WORD	.S.	
020274	000042	BZ	138	
020274	101456	NICPC=NICPC+1		
020276	000043	.WORD	.S.	
020276	061224	MOVE	BREG,OUT1 <CSR4> ;ELSE, REPORT ERROR...	
020300	000044	NICPC=NICPC+1		
020300	060600	.WORD	.S.	
020300		MOVE	SPAD <0>,BREG ;	
020300		NICPC=NICPC+1		
020300		.WORD	.S.	

5604	020302	000045	MOVE BREG OUT1 (CSRS)	;BAD DATA...
5605			MICPC=MICPC+1	
5606	020302	061225	.WORD .S.	
5607	020304		MOVE # 4,BREG	;TYPE OF ERROR...
5608		000046	MICPC=MICPC+1	
5609	020304	000404	.WORD .S.	
5610	020306		MOVE BREG OUT1 (CSR3)	;
5611		000047	MICPC=MICPC+1	
5612	020306	061223	.WORD .S.	
5613	020310		MOVE # 5,BREG	;LOAD ADDRESS.
5614		000050	MICPC=MICPC+1	
5615	020310	000405	.WORD .S.	
5616	020312		MOVE BREG OUT1 (CSR7)	;
5617		000051	MICPC=MICPC+1	
5618	020312	061227	.WORD .S.	
5619	020314		CALL EROR	;DATA ERROR!!
5620	020314		MOVE # (MICPC+3),BREG	
5621		000052	MICPC=MICPC+1	
5622	020314	000454	.WORD .S.	
5623	020316		SBR EROR	
5624		000053	MICPC=MICPC+1	
5625	020316	104400	.WORD .S.	
5626	020320		MOVE MEM,BREG	;RESTORE BREG...
5627		000054	MICPC=MICPC+1	
5628	020320	040620	.WORD .S.	
5629	020322		SBR 125	;LOOP ON ERROR...
5630		000055	MICPC=MICPC+1	
5631	020322	100435	.WORD .S.	
5632	020324		CALL SCP1	;IS LOOP DATA SET???
5633	020324		MOVE # (MICPC+3),BREG	
5634		000056	MICPC=MICPC+1	
5635	020324	000460	.WORD .S.	
5636	020326		SBR SCP1	
5637		000057	MICPC=MICPC+1	
5638	020326	104427	.WORD .S.	
5639	020330		MOVE MEM,BREG	;RESTORE BREG...
5640		000060	MICPC=MICPC+1	
5641	020330	040620	.WORD .S.	
5642	020332		SBR 125	;LOOP ON DATA.
5643		000061	MICPC=MICPC+1	
5644	020332	100435	.WORD .S.	
5645	020334		MOVE MEM,BREG	;RESTORE BREG...
5646		000062	MICPC=MICPC+1	
5647	020334	040620	.WORD .S.	
5648	020336		SHFBRT	;SET THE NEXT BIT...
5649		000063	MICPC=MICPC+1	
5650	020336	061620	.WORD SBR!.SELB!.DBRSH	
5651	020340		BB7 125	;BRANCH IF NOT DONE...
5652		000064	MICPC=MICPC+1	
5653	020340	103435	.WORD .S.	
5654	020342		CALL SCPE	
5655	020342		MOVE # (MICPC+3),BREG	
5656		000065	MICPC=MICPC+1	
5657	020342	000467	.WORD .S.	
5658	020344		SBR SCPE	
5659	020344			

135:

145:

```

5660      000066      MICPC=MICPC+1
5661 020344 104454      .WORD .S.
5662 020346      SBR 218
5663      000067      MICPC=MICPC+1
5664 020346 170400      .WORD .S.
5665 020350      SSPTS1 6
5666 020350      SXZ
5667
5668
5669      ;***** TEST 22 *****
5670      ;* SCRATCH PAD TEST FOR SP6
5671      ;* FLOAT A 1 THROUGH SCRATCH PAD 6
5672      ;* FLOAT A 0 THROUGH SCRATCH PAD 6
5673 020350      SXZ
5674      ;*****
5675
5676 020350      STSTN
5677
5678      ; TEST 22
5679 020350 012737 000022 001202 TST22: MOV #22,STSTNM ; LOAD THE NO. OF THIS TEST
5680 020356 012737 020564 001442 MOV #TST23,NEXT ; POINT TO THE START OF NEXT TEST.
5681      ;R1 CONTAINS BASE KMC11 ADDRESS
5682 020364 004737 035536 JSR PC,LDRMT ;LOAD-VERIFY-WAIT.
5683 020370 020404 MCT22
5684 020372 104022 ERROR 22 ;TIME OUT ERROR...
5685 020374 012706 001200 MOV #STACK,SP ;RESET STACK.
5686 020400 000177 161036 JMP #NEXT ;GO TO NEXT TEST...
5687
5688      MCT22:
5689      218:
5690      MOVE #0,BREG ;SET TO CLEAR SPAD 16...
5691 020404 000000 MICPC=MICPC+1
5692 020406 000400 .WORD .S.
5693      MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES.
5694      MICPC=MICPC+1
5695      .WORD .S.
5696 020410 000001 SSPFLT 1,18,25,38,45,6
5697 020410 063236 18: MOVE #200,BREG ;START WITH BIT 7.
5698      MICPC=MICPC+1
5699      .WORD .S.
5700      25: MOVE BREG,SPAD <6> ;LOAD THE SCRATCH PAD...
5701 020412 000003 MICPC=MICPC+1
5702 020414 063226 .WORD .S.
5703      MOVE SPAD <6>,MEM ;GET THE "FOUND"...
5704 020414 000004 MICPC=MICPC+1
5705 020416 062606 .WORD .S.
5706      MOVE MEM,SPAD <0> ;
5707 020416 043220 MICPC=MICPC+1
5708 020420 .WORD .S.
5709      MOVE BREG,MEM ;SAVE THE CONTENTS OF BREG...
5710 020420 000006 MICPC=MICPC+1
5711 020422 062620 .WORD .S.
5712      SIFEQ BREG,SPAD <0> 3S ;IF GOOD.. CONTINUE...
5713
5714 020422      SUB2C SPAD <0>,BREG,NOP
5715      000007 MICPC=MICPC+1

```

5716	020422	060360	.WORD .S.	
5717	020424		BZ .S.	
5718		000010	NICPC=NICPC+1	
5719	020424	101424	.WORD .S.	
5720	020426		MOVE BREG OUT1 <CSR4>	;ELSE, REPORT ERROR...
5721		000011	NICPC=NICPC+1	
5722	020426	061224	.WORD .S.	
5723	020430		MOVE SPAD <0>, BREG	;
5724		000012	NICPC=NICPC+1	
5725	020430	060600	.WORD .S.	
5726	020432		MOVE BREG OUT1 <CSR5>	;BAD DATA...
5727		000013	NICPC=NICPC+1	
5728	020432	061225	.WORD .S.	
5729	020434		MOVE B 4 BREG	;TYPE OF ERROR...
5730		000014	NICPC=NICPC+1	
5731	020434	000404	.WORD .S.	
5732	020436		MOVE BREG OUT1 <CSR3>	;
5733		000015	NICPC=NICPC+1	
5734	020436	061223	.WORD .S.	
5735	020440		MOVE B 6 BREG	;LOAD ADDRESS.
5736		000016	NICPC=NICPC+1	
5737	020440	000406	.WORD .S.	
5738	020442		MOVE BREG OUT1 <CSR7>	;
5739		000017	NICPC=NICPC+1	
5740	020442	061227	.WORD .S.	
5741	020444		CALL EROR	;DATA ERROR!!
5742	020444		MOVE B <NICPC+3>, BREG	
5743		000020	NICPC=NICPC+1	
5744	020444	000422	.WORD .S.	
5745	020446		SBR EROR	
5746		000021	NICPC=NICPC+1	
5747	020446	104400	.WORD .S.	
5748	020450		MOVE NEH BREG	;RESTORE BREG...
5749		000022	NICPC=NICPC+1	
5750	020450	040620	.WORD .S.	
5751	020452		SBR	;LOOP ON ERROR...
5752		000023	NICPC=NICPC+1	
5753	020452	100403	.WORD .S.	
5754	020454		CALL SCP1	;IS LOOP DATA SET???
5755	020454		MOVE B <NICPC+3>, BREG	
5756		000024	NICPC=NICPC+1	
5757	020454	000426	.WORD .S.	
5758	020456		SBR SCP1	
5759		000025	NICPC=NICPC+1	
5760	020456	104427	.WORD .S.	
5761	020460		MOVE NEH BREG	;RESTORE BREG...
5762		000026	NICPC=NICPC+1	
5763	020460	040620	.WORD .S.	
5764	020462		SBR	;LOOP ON DATA.
5765		000027	NICPC=NICPC+1	
5766	020462	100403	.WORD .S.	
5767	020464		MOVE NEH BREG	;RESTORE BREG...
5768		000030	NICPC=NICPC+1	
5769	020464	040620	.WORD .S.	
5770	020466		SFBRT	;SET THE NEXT BIT...
5771		000031	NICPC=NICPC+1	

35:

DZKCA MACY11 27(1006) 12-MAY-77 14:07 PAGE 110
DZKCA.P11 12-MAY-77 13:58

KMC11 SCRATCH PAD TESTS

5778	020466	061620	.WORD	SBR!..SELB!..DBRSH	
5779	020470		BZ	45	;BRANCH IF DONE...
5779	020470	000032	NICPC=NICPC+1		
5779	020472	100434	.WORD	5	
5779	020472		SBR	23	;CONTINUE...
5779	020472	000033	NICPC=NICPC+1		
5779	020472	100403	.WORD	5	
5779	020474				
5779	020474				
5779	020474				
5779	020474				
5779	020474	000034	SPFLT	0,115,125,135,145,6	
5779	020474	000037	115:	MOVE #177,BREG ;START WITH BIT 7.	
5779	020474		NICPC=NICPC+1		
5779	020476	000037	125:	MOVE #BREG,SPAD <6> ;LOAD THE SCRATCH PAD...	
5779	020476	000036	NICPC=NICPC+1		
5779	020476	062225	.WORD	5	
5779	020476		MOVE #SPAD <6>,MEM ;GET THE "FOUND"...		
5779	020476		NICPC=NICPC+1		
5779	020476	000036	.WORD	5	
5779	020476	062226	MOVE #MEM,SPAD <0>		;
5779	020476		NICPC=NICPC+1		
5779	020476	000037	.WORD	5	
5779	020476	062220	MOVE #BREG,MEM		;SAVE THE CONTENTS OF BREG...
5779	020476		NICPC=NICPC+1		
5779	020476	000040	.WORD	5	
5779	020476	062220	SIFEQ #BREG,SPAD <0> 135		;IF GOOD.. CONTINUE...
5779	020476				
5779	020506		SUB2C #SPAD <0>,BREG,NOP		
5779	020506	000041	NICPC=NICPC+1		
5779	020506	060360	.WORD	5	
5779	020510		BZ	135	
5779	020510	000042	NICPC=NICPC+1		
5779	020510	101456	.WORD	5	
5779	020512		MOVE #BREG,OUT1 <CSR4>		;ELSE, REPORT ERROR...
5779	020512	000043	NICPC=NICPC+1		
5779	020512	061224	.WORD	5	
5779	020514		MOVE #SPAD <0>,BREG		;
5779	020514	000044	NICPC=NICPC+1		
5779	020514	060600	.WORD	5	
5779	020516		MOVE #BREG,OUT1 <CSR5>		;BAD DATA...
5779	020516	000045	NICPC=NICPC+1		
5779	020516	061225	.WORD	5	
5779	020520		MOVE #4,BREG		;TYPE OF ERROR...
5779	020520	000046	NICPC=NICPC+1		
5779	020520	000404	.WORD	5	
5779	020522		MOVE #BREG,OUT1 <CSR3>		;
5779	020522	000047	NICPC=NICPC+1		
5779	020522	061223	.WORD	5	
5779	020524		MOVE #6,BREG		;LOAD ADDRESS.
5779	020524	000050	NICPC=NICPC+1		
5779	020524	000406	.WORD	5	
5779	020526		MOVE #BREG,OUT1 <CSR7>		;
5779	020526	000051	NICPC=NICPC+1		
5779	020526	061227	.WORD	5	
5779	020530		CALL ERROR		;DATA ERROR!!
5779	020530		MOVE # <MICPC+3>,BREG		

5868
5869
5870
5871
5872
5873
5874
5875
5876
5877
5878
5879
5880
5881
5882
5883

020530 000052
020532 000454
020538 000053
020540 104400
020544 000054
020546 040620
020550 000055
020552 100435
020554 000056
020556 000460
020560 000057
020562 104427
020564 000060
020566 040620
020568 000061
020570 100435
020572 000062
020574 040620
020576 000063
020578 061620
020580 000064
020582 103435
020584 000065
020586 000467
020588 000066
020590 104454
020592 000067
020594 100400
020596 020564
020598 020564
020600 020564
020602 020564

```
NICPC=NICPC+1
.WORD .S.
SBR ERROR
NICPC=NICPC+1
.WORD .S.
MOVE MEM,BREG ;RESTORE BREG...
NICPC=NICPC+1
.WORD .S.
SBR 125 ;LOOP ON ERROR...
135: CALL SCP1 ;IS LOOP DATA SET???
MOVE 8<NICPC+3>,BREG
NICPC=NICPC+1
.WORD .S.
SBR SCP1
NICPC=NICPC+1
.WORD .S.
MOVE MEM,BREG ;RESTORE BREG...
NICPC=NICPC+1
.WORD .S.
SBR 125 ;LOOP ON DATA.
MOVE MEM,BREG ;RESTORE BREG...
NICPC=NICPC+1
.WORD .S.
SHFBT ;SET THE NEXT BIT...
NICPC=NICPC+1
.WORD SBR!..SELB!..DBRSH
BB7 125 ;BRANCH IF NOT DONE...
145: CALL SCPE
MOVE 8<NICPC+3>,BREG
NICPC=NICPC+1
.WORD .S.
SBR SCPE
NICPC=NICPC+1
.WORD .S.
SBR 218
NICPC=NICPC+1
.WORD .S.
SSPTS1 7
SXZ

;***** TEST 23 *****
;# SCRATCH PAD TEST FOR SP7
;# FLOAT A 1 THROUGH SCRATCH PAD 7
;# FLOAT A 0 THROUGH SCRATCH PAD 7
SXZ

;*****
```

SXZ
STSTN

; TEST 23

5884	020514	012737	000023	001202	TST23:	MOV	#23,STSTNM		
5885	020514	012737	021000	001442		MOV	#STST24,NEXT		; LOAD THE NO. OF THIS TEST
5886	020514	012737	021000	001442					; POINT TO THE START OF NEXT TEST.
5887	020514	012737	021000	001442					;R1 CONTAINS BASE KMC11 ADDRESS
5888	020514	012737	021000	001442					;LOAD-VERIFY-WAIT.
5889	020514	012737	021000	001442					
5890	020514	012737	021000	001442					
5891	020514	012737	021000	001442					
5892	020514	012737	021000	001442					
5893	020514	012737	021000	001442					
5894	020514	012737	021000	001442					
5895	020514	012737	021000	001442					
5896	020514	012737	021000	001442					
5897	020514	012737	021000	001442					
5898	020514	012737	021000	001442					
5899	020514	012737	021000	001442					
5900	020514	012737	021000	001442					
5901	020514	012737	021000	001442					
5902	020514	012737	021000	001442					
5903	020514	012737	021000	001442					
5904	020514	012737	021000	001442					
5905	020514	012737	021000	001442					
5906	020514	012737	021000	001442					
5907	020514	012737	021000	001442					
5908	020514	012737	021000	001442					
5909	020514	012737	021000	001442					
5910	020514	012737	021000	001442					
5911	020514	012737	021000	001442					
5912	020514	012737	021000	001442					
5913	020514	012737	021000	001442					
5914	020514	012737	021000	001442					
5915	020514	012737	021000	001442					
5916	020514	012737	021000	001442					
5917	020514	012737	021000	001442					
5918	020514	012737	021000	001442					
5919	020514	012737	021000	001442					
5920	020514	012737	021000	001442					
5921	020514	012737	021000	001442					
5922	020514	012737	021000	001442					
5923	020514	012737	021000	001442					
5924	020514	012737	021000	001442					
5925	020514	012737	021000	001442					
5926	020514	012737	021000	001442					
5927	020514	012737	021000	001442					
5928	020514	012737	021000	001442					
5929	020514	012737	021000	001442					
5930	020514	012737	021000	001442					
5931	020514	012737	021000	001442					
5932	020514	012737	021000	001442					
5933	020514	012737	021000	001442					
5934	020514	012737	021000	001442					
5935	020514	012737	021000	001442					
5936	020514	012737	021000	001442					
5937	020514	012737	021000	001442					
5938	020514	012737	021000	001442					
5939	020514	012737	021000	001442					

```

; LOAD THE NO. OF THIS TEST
; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
;LOAD-VERIFY-WAIT.

;TIME OUT ERROR...
;RESET STACK...
;GO TO NEXT TEST...

```

```

;SET TO CLEAR SPAD 16...
;FOR RETURN ADDRESS PURPOSES.

```

```

;START WITH BIT 7.

```

```

;LOAD THE SCRATCH PAD...

```

```

;GET THE "FOUND"...

```

```

;SAVE THE CONTENTS OF BREG...

```

```

;IF GOOD.. CONTINUE...

```

```

;ELSE, REPORT ERROR...

```

```

;BAD DATA...

```

```

;TYPE OF ERROR...

```

5940		000015	MICPC=MICPC+1	
5941	020652	061223	.WORD .S.	
5942	020654		MOVE # 7, BREG	;LOAD ADDRESS.
5943		000016	MICPC=MICPC+1	
5944	020654	000407	.WORD .S.	
5945	020656		MOVE BREG, OUT1 (CSR7)	;
5946		000017	MICPC=MICPC+1	
5947	020656	061227	.WORD .S.	
5948	020660		CALL EROR	;DATA ERROR!!
5949	020660		MOVE # <MICPC+3>, BREG	
5950		000020	MICPC=MICPC+1	
5951	020660	000422	.WORD .S.	
5952	020662		SBR EROR	
5953		000021	MICPC=MICPC+1	
5954	020662	104400	.WORD .S.	
5955	020664		MOVE MEM, BREG	;RESTORE BREG...
5956		000022	MICPC=MICPC+1	
5957	020664	040620	.WORD .S.	
5958	020666		SBR 25	;LOOP ON ERROR...
5959		000023	MICPC=MICPC+1	
5960	020666	100403	.WORD .S.	
5961	020670		CALL SCP1	;IS LOOP DATA SET???
5962	020670		MOVE # <MICPC+3>, BREG	
5963		000024	MICPC=MICPC+1	
5964	020670	000426	.WORD .S.	
5965	020672		SBR SCP1	
5966		000025	MICPC=MICPC+1	
5967	020672	104427	.WORD .S.	
5968	020674		MOVE MEM, BREG	;RESTORE BREG...
5969		000026	MICPC=MICPC+1	
5970	020674	040620	.WORD .S.	
5971	020676		SBR 25	;LOOP ON DATA.
5972		000027	MICPC=MICPC+1	
5973	020676	100403	.WORD .S.	
5974	020700		MOVE MEM, BREG	;RESTORE BREG...
5975		000030	MICPC=MICPC+1	
5976	020700	040620	.WORD .S.	
5977	020702		SHFBT	;SET THE NEXT BIT...
5978		000031	MICPC=MICPC+1	
5979	020702	061620	.WORD SBR!.SELB!.DBRSH	
5980	020704		BB7 45	;BRANCH IF DONE...
5981		000032	MICPC=MICPC+1	
5982	020704	103434	.WORD .S.	
5983	020706		SBR 25	;CONTINUE...
5984		000033	MICPC=MICPC+1	
5985	020706	100403	.WORD .S.	
5986	020710			
5987	020710			
5988	020710			
5989		000034		
5990	020710	000577	SSPFLT 0, 115, 125, 135, 145, 7	
5991	020712		115: MOVE # 177, BREG ;START WITH BIT 7.	
5992		000035	MICPC=MICPC+1	
5993	020712	063227	.WORD .S.	
5994	020714		125: MOVE BREG SPAD <7> ;LOAD THE SCRATCH PAD...	
5995		000036	MICPC=MICPC+1	
			.WORD .S.	
			MOVE SPAD <7>, MEM ;GET THE "FOUND"...	
			MICPC=MICPC+1	

5996	020714	062607	.WORD .S.	
5997	020716		MOVE MEM,SPAD <0>	;
5998		000037	MICPC=MICPC+1	
5999	020716	043220	.WORD .S.	
6000	020720		MOVE BREG, MEM	;SAVE THE CONTENTS OF BREG...
6001		000040	MICPC=MICPC+1	
6002	020720	062620	.WORD .S.	
6003	020722		SIFEQ BREG,SPAD <0> 13\$;IF GOOD.. CONTINUE...
6004				
6005				
6006	020722		SUB2C SPAD <0>,BREG,NOP	
6007		000041	MICPC=MICPC+1	
6008	020722	060360	.WORD .S.	
6009	020724		BZ 13\$	
6010		000042	MICPC=MICPC+1	
6011	020724	101456	.WORD .S.	
6012	020726		MOVE BREG,OUT1 <CSR4>	;ELSE, REPORT ERROR...
6013		000043	MICPC=MICPC+1	
6014	020726	061224	.WORD .S.	
6015	020730		MOVE SPAD <0>,BREG	;
6016		000044	MICPC=MICPC+1	
6017	020730	060600	.WORD .S.	
6018	020732		MOVE BREG,OUT1 <CSR5>	;BAD DATA...
6019		000045	MICPC=MICPC+1	
6020	020732	061225	.WORD .S.	
6021	020734		MOVE # 4,BREG	;TYPE OF ERROR...
6022		000046	MICPC=MICPC+1	
6023	020734	000404	.WORD .S.	
6024	020736		MOVE BREG,OUT1 <CSR3>	;
6025		000047	MICPC=MICPC+1	
6026	020736	061223	.WORD .S.	
6027	020740		MOVE # 7,BREG	;LOAD ADDRESS.
6028		000050	MICPC=MICPC+1	
6029	020740	000407	.WORD .S.	
6030	020742		MOVE BREG,OUT1 <CSR7>	;
6031		000051	MICPC=MICPC+1	
6032	020742	061227	.WORD .S.	
6033	020744		CALL EROR	;DATA ERROR!!
6034	020744		MOVE # <MICPC+3>,BREG	
6035		000052	MICPC=MICPC+1	
6036	020744	000454	.WORD .S.	
6037	020746		SBR EROR	
6038		000053	MICPC=MICPC+1	
6039	020746	104400	.WORD .S.	
6040	020750		MOVE MEM,BREG	;RESTORE BREG...
6041		000054	MICPC=MICPC+1	
6042	020750	040620	.WORD .S.	
6043	020752		SBR 12\$;LOOP ON ERROR...
6044		000055	MICPC=MICPC+1	
6045	020752	100435	.WORD .S.	
6046	020754		CALL SCP1	;IS LOOP DATA SET???
6047	020754		MOVE # <MICPC+3>,BREG	
6048		000056	MICPC=MICPC+1	
6049	020754	060460	.WORD .S.	
6050	020756		SBR SCP1	
6051		000057	MICPC=MICPC+1	

13\$:

6052	020756	104427		
6053	020760			
6054		000060		
6055	020760	040620		
6056	020762			
6057		000061		
6058	020762	100435		
6059	020764			
6060		000062		
6061	020764	040620		
6062	020766			
6063		000063		
6064	020766	061620		
6065	020770			
6066		000064		
6067	020770	103435		
6068	020772			
6069	020772			
6070	020772			
6071		000065		
6072	020772	000467		
6073	020774			
6074		000066		
6075	020774	104454		
6076	020776			
6077		000067		
6078	020776	100400		
6079	021000			
6080	021000			
6081				
6082				
6083				
6084				
6085				
6086				
6087	021000			
6088				
6089				
6090	021000			
6091				
6092				
6093	021000	012737	000024	001202
6094	021006	012737	021214	001442
6095				
6096	021014	004737	035536	
6097	021020	021034		
6098	021022	104022		
6099	021024	012706	001200	
6100	021030	000177	160406	
6101	021034			
6102	021034			
6103	021034			
6104		000000		
6105	021034	000400		
6106	021036			
6107		000001		

```

.WORD .S.
MOVE MEM, BREG ;RESTORE BREG...
NICPC=NICPC+1
.WORD .S.
SBR 128 ;LOOP ON DATA.
NICPC=NICPC+1
.WORD .S.
MOVE MEM, BREG ;RESTORE BREG...
NICPC=NICPC+1
.WORD .S.
SHFBRT ;SET THE NEXT BIT...
NICPC=NICPC+1
.WORD .SBR!..SELB!..DBRSH
BB7 128 ;BRANCH IF NOT DONE...
NICPC=NICPC+1
.WORD .S.
145:
CALL SCPE
MOVE # (NICPC+3), BREG
NICPC=NICPC+1
.WORD .S.
SBR SCPE
NICPC=NICPC+1
.WORD .S.
SBR 218
NICPC=NICPC+1
.WORD .S.
SSPTS1 10
SXZ

;***** TEST 24 *****
;* SCRATCH PAD TEST FOR SP10
;* FLOAT A 1 THROUGH SCRATCH PAD 10
;* FLOAT A 0 THROUGH SCRATCH PAD 10
SXZ
;*****

STSTN
; TEST 24
-----
TST24: MOV #24, STSTNM ; LOAD THE NO. OF THIS TEST
MOV #TST25, NEXT ; POINT TO THE START OF NEXT TEST.
; R1 CONTAINS BASE KMC11 ADDRESS
; LOAD-VERIFY-WAIT.
JSR PC, LDVRWT
MCT24
ERROR 22 ; TIME OUT ERROR...
MOV #STACK, SP ; RESET STACK.
JMP @NEXT ; GO TO NEXT TEST...

MCT24:
218:
MOVE # 0, BREG ; SET TO CLEAR SPAD 16...
NICPC=NICPC+1
.WORD .S.
MOVE BREG, SPAD (16) ; FOR RETURN ADDRESS PURPOSES.
NICPC=NICPC+1

```

KMC11 SCRATCH PAD TESTS

0442

6108	021036	063236	
6109	021040		
6110	021040		
6111		000002	
6112	021040	000600	
6113	021042		
6114		000003	
6115	021042	063230	
6116	021044		
6117		000004	
6118	021044	062610	
6119	021046		
6120		000005	
6121	021046	043220	
6122	021050		
6123		000006	
6124	021050	062620	
6125	021052		
6126			
6127			
6128	021052		
6129		000007	
6130	021052	060360	
6131	021054		
6132		000010	
6133	021054	101424	
6134	021056		
6135		000011	
6136	021056	061224	
6137	021060		
6138		000012	
6139	021060	060600	
6140	021062		
6141		000013	
6142	021062	061225	
6143	021064		
6144		000014	
6145	021064	000404	
6146	021066		
6147		000015	
6148	021066	061223	
6149	021070		
6150		000016	
6151	021070	000410	
6152	021072		
6153		000017	
6154	021072	061227	
6155	021074		
6156	021074		
6157		000020	
6158	021074	000422	
6159	021076		
6160		000021	
6161	021076	104400	
6162	021100		
6163		000022	

```

      .WORD      .S.
SSPFLT 1, 18, 25, 38, 45, 10
15:    MOVE      # 200, BREG ; START WITH BIT 7.
      .WORD      .S.
      MICPC=MICPC+1
25:    .WORD      .S.
      MOVE      BREG, SPAD <10> ; LOAD THE SCRATCH PAD...
      .WORD      .S.
      MICPC=MICPC+1
      .WORD      .S.
      MOVE      SPAD <10>, MEM ; GET THE "FOUND"...
      .WORD      .S.
      MICPC=MICPC+1
      .WORD      .S.
      MOVE      MEM, SPAD <0> ;
      .WORD      .S.
      MICPC=MICPC+1
      .WORD      .S.
      MOVE      BREG, MEM ; SAVE THE CONTENTS OF BREG...
      .WORD      .S.
      MICPC=MICPC+1
      .WORD      .S.
      SIFEQ     BREG, SPAD <0> 35 ; IF GOOD.. CONTINUE...

      .WORD      .S.
SUB2C   SPAD <0>, BREG, NOP
      .WORD      .S.
      MICPC=MICPC+1
      .WORD      .S.
      BZ        35
      .WORD      .S.
      MICPC=MICPC+1
      .WORD      .S.
      MOVE      BREG, OUT1 <CSR4> ; ELSE, REPORT ERROR...
      .WORD      .S.
      MICPC=MICPC+1
      .WORD      .S.
      MOVE      SPAD <0>, BREG ;
      .WORD      .S.
      MICPC=MICPC+1
      .WORD      .S.
      MOVE      BREG, OUT1 <CSR5> ; BAD DATA...
      .WORD      .S.
      MICPC=MICPC+1
      .WORD      .S.
      MOVE      # 4, BREG ; TYPE OF ERROR...
      .WORD      .S.
      MICPC=MICPC+1
      .WORD      .S.
      MOVE      BREG, OUT1 <CSR3> ;
      .WORD      .S.
      MICPC=MICPC+1
      .WORD      .S.
      MOVE      # 10, BREG ; LOAD ADDRESS.
      .WORD      .S.
      MICPC=MICPC+1
      .WORD      .S.
      MOVE      BREG, OUT1 <CSR7> ;
      .WORD      .S.
      MICPC=MICPC+1
      .WORD      .S.
      CALL     EROR ; DATA ERROR!!
      .WORD      .S.
      MOVE      # <MICPC+3>, BREG
      .WORD      .S.
      MICPC=MICPC+1
      .WORD      .S.
      SBR      EROR
      .WORD      .S.
      MICPC=MICPC+1
      .WORD      .S.
      MOVE      MEM, BREG ; RESTORE BREG...
      .WORD      .S.
      MICPC=MICPC+1

```

6164	021100	040620	.WORD .S.	
6165	021102		SBR 25	;LOOP ON ERROR...
6166		000023	MICPC=MICPC+1	
6167	021102	100403	.WORD .S.	
6168	021104		35: CALL SCP1	;IS LOOP DATA SET???
6169	021104		MOVE 8 <MICPC+3>,BREG	
6170		000024	MICPC=MICPC+1	
6171	021104	000426	.WORD .S.	
6172	021106		SBR SCP1	
6173		000025	MICPC=MICPC+1	
6174	021106	104427	.WORD .S.	
6175	021110		MOVE MEM,BREG	;RESTORE BREG...
6176		000026	MICPC=MICPC+1	
6177	021110	040620	.WORD .S.	
6178	021112		SBR 25	;LOOP ON DATA.
6179		000027	MICPC=MICPC+1	
6180	021112	100403	.WORD .S.	
6181	021114		MOVE MEM,BREG	;RESTORE BREG...
6182		000030	MICPC=MICPC+1	
6183	021114	040620	.WORD .S.	
6184	021116		SHFBT	;SET THE NEXT BIT...
6185		000031	MICPC=MICPC+1	
6186	021116	061620	.WORD SBR!.SELB!.DBRSH	
6187	021120		BZ 45	;BRANCH IF DONE...
6188		000032	MICPC=MICPC+1	
6189	021120	103434	.WORD .S.	
6190	021122		SBR 25	;CONTINUE...
6191		000033	MICPC=MICPC+1	
6192	021122	100403	.WORD .S.	
6193	021124		45: SSPFLT 0,115,125,135,145,10	
6194	021124		115: MOVE 8,177,BREG ;START WITH BIT 7.	
6195	021124		MICPC=MICPC+1	
6196		000034	.WORD .S.	
6197	021124	000577	125: MOVE BREG,SPAD <10>	;LOAD THE SCRATCH PAD...
6198	021126		MICPC=MICPC+1	
6199		000035	.WORD .S.	
6200	021126	063230	MOVE SPAD <10>,MEM	;GET THE "FOUND"...
6201	021130		MICPC=MICPC+1	
6202		000036	.WORD .S.	
6203	021130	062610	MOVE MEM,SPAD <0>	;
6204	021132		MICPC=MICPC+1	
6205		000037	.WORD .S.	
6206	021132	043220	MOVE BREG,MEM	;SAVE THE CONTENTS OF BREG...
6207	021134		MICPC=MICPC+1	
6208		000040	.WORD .S.	
6209	021134	062620	\$IFEQ BREG,SPAD <0> 135	;IF GOOD.. CONTINUE...
6210	021136			
6211				
6212				
6213	021136		SUB2C SPAD <0>,BREG,NOP	
6214		000041	MICPC=MICPC+1	
6215	021136	060360	.WORD .S.	
6216	021140		BZ 135	
6217		000042	MICPC=MICPC+1	
6218	021140	101456	.WORD .S.	
6219	021142		MOVE BREG,OUT1 <CSR4>	;ELSE, REPORT ERROR...

6223	021142	000043	MICPC=MICPC+1	
6224	021144	061224	.WORD .S.	
6225	021144	000044	MOVE SPAD (0),BREG	;
6226	021146	060600	MICPC=MICPC+1	
6227	021146	000045	.WORD .S.	
6228	021150	061225	MOVE BREG OUT1 (CSRS)	;BAD DATA...
6229	021150	000046	MICPC=MICPC+1	
6230	021152	000404	.WORD .S.	
6231	021152	000404	MOVE BREG OUT1 (CSR3)	;
6232	021152	000047	MICPC=MICPC+1	
6233	021154	061223	.WORD .S.	
6234	021154	000050	MOVE # 4,BREG	;TYPE OF ERROR...
6235	021154	000410	MICPC=MICPC+1	
6236	021154	000410	.WORD .S.	
6237	021156	000051	MOVE BREG OUT1 (CSR7)	;
6238	021156	061227	MICPC=MICPC+1	
6239	021160	000052	.WORD .S.	
6240	021160	000454	CALL EROR	;DATA ERROR!!
6241	021162	000053	MOVE # (MICPC+3),BREG	
6242	021162	104400	MICPC=MICPC+1	
6243	021164	000054	.WORD .S.	
6244	021164	040620	SBR EROR	
6245	021166	000055	MICPC=MICPC+1	
6246	021166	100435	.WORD .S.	
6247	021170	000056	CALL SCP1	135: ;IS LOOP DATA SET???
6248	021170	000460	MOVE # (MICPC+3),BREG	
6249	021172	000057	MICPC=MICPC+1	
6250	021172	104427	.WORD .S.	
6251	021174	000060	SBR SCP1	
6252	021174	040620	MICPC=MICPC+1	
6253	021176	000061	.WORD .S.	
6254	021200	000062	MOVE MEM,BREG	;RESTORE BREG...
6255	021200	040620	MICPC=MICPC+1	
6256	021202	000063	.WORD .S.	
6257	021202	061620	SHFBRT	;SET THE NEXT BIT...
6258	021204	000064	MICPC=MICPC+1	
6259	021204	103435	.WORD .S.	
6260	021206		BB7 125 ;BRANCH IF NOT DONE...	
6261	021206			
6262	021206			
6263	021206			
6264	021206			
6265	021206			
6266	021206			
6267	021206			
6268	021206			
6269	021206			
6270	021206			
6271	021206			
6272	021206			
6273	021206			
6274	021206			
6275	021206			

145:

KMC11 SCRATCH PAD TESTS

```

6276 021206 CALL SCPE
6277 021206 MOVE # <MICPC+3>, BREG
6278 000065 MICPC=MICPC+1
6279 021206 000467 .WORD .S.
6280 021210 JSR SCPE
6281 000066 MICPC=MICPC+1
6282 021210 104454 .WORD .S.
6283 021212 JSR 21$
6284 000067 MICPC=MICPC+1
6285 021212 100400 .WORD .S.
6286 021214 $SPTS1 11
6287 021214 SXZ
6288
6289
6290
6291
6292
6293
6294
6295
6296
6297
6298
6299
6300
6301
6302
6303
6304
6305
6306
6307
6308
6309
6310
6311
6312
6313
6314
6315
6316
6317
6318
6319
6320
6321
6322
6323
6324
6325
6326
6327
6328
6329
6330
6331

```

;***** TEST 25 *****
; SCRATCH PAD TEST FOR SP11
; FLOAT A 1 THROUGH SCRATCH PAD 11
; FLOAT A 0 THROUGH SCRATCH PAD 11
;*****

```

021214 SXZ
021214 STSTN
; TEST 25
-----
021214 012737 000025 001202 TST25: MOV #25, STSTNM ; LOAD THE NO. OF THIS TEST
021222 012737 021430 001442 MOV #TST26, NEXT ; POINT TO THE START OF NEXT TEST.
; R1 CONTAINS BASE KMC11 ADDRESS
; LOAD-VERIFY-WAIT.
021230 004737 035536 JSR PC, LDVRMT
021250 021250 MCT25:
104022 ERROR 22 ; TIME OUT ERROR...
012706 001200 MOV #STACK, SP ; RESET STACK.
000177 160172 JMP @NEXT ; GO TO NEXT TEST...
MCT25:
21$: MOVE # 0, BREG ; SET TO CLEAR SPAD 16...
MICPC=MICPC+1
.WORD .S.
MOVE BREG, SPAD <16> ; FOR RETURN ADDRESS PURPOSES.
MICPC=MICPC+1
.WORD .S.
SSPFLT 1, 15, 25, 35, 45, 11
15: MOVE # 200, BREG ; START WITH BIT 7.
MICPC=MICPC+1
.WORD .S.
25: MOVE BREG, SPAD <11> ; LOAD THE SCRATCH PAD...
MICPC=MICPC+1
.WORD .S.
MOVE SPAD <11>, MEM ; GET THE "FOUND"...
MICPC=MICPC+1
.WORD .S.
MOVE MEM, SPAD <0> ;
MICPC=MICPC+1
.WORD .S.
MOVE BREG, MEM ; SAVE THE CONTENTS OF BREG...
MICPC=MICPC+1
.WORD .S.

```

6332	021266		SIFE0 BREG,SPAD <0> 35	;IF GOOD.. CONTINUE...
6333				
6335	021266		SUB2C SPAD <0>,BREG,NOP	
6336		000007	MICPC=MICPC+1	
6337	021266	060360	.WORD .S.	
6338	021270			
6339		000010	MICPC=MICPC+1	
6340	021270	101424	.WORD .S.	
6341	021272		MOVE BREG,OUT1 <CSR4>	;ELSE, REPORT ERROR...
6342		000011	MICPC=MICPC+1	
6343	021272	061224	.WORD .S.	
6344	021274		MOVE SPAD <0>,BREG	;
6345		000012	MICPC=MICPC+1	
6346	021274	060600	.WORD .S.	
6347	021276		MOVE BREG,OUT1 <CSR5>	;BAD DATA...
6348		000013	MICPC=MICPC+1	
6349	021276	061225	.WORD .S.	
6350	021300		MOVE # 4 BREG	;TYPE OF ERROR...
6351		000014	MICPC=MICPC+1	
6352	021300	000404	.WORD .S.	
6353	021302		MOVE BREG,OUT1 <CSR3>	;
6354		000015	MICPC=MICPC+1	
6355	021302	061223	.WORD .S.	
6356	021304		MOVE # 11 BREG	;LOAD ADDRESS.
6357		000016	MICPC=MICPC+1	
6358	021304	000411	.WORD .S.	
6359	021306		MOVE BREG,OUT1 <CSR7>	;
6360		000017	MICPC=MICPC+1	
6361	021306	061227	.WORD .S.	
6362	021310		CALL EROR	;DATA ERROR!!
6363	021310		MOVE # <MICPC+3>,BREG	
6364		000020	MICPC=MICPC+1	
6365	021310	000422	.WORD .S.	
6366	021312		SBR EROR	
6367		000021	MICPC=MICPC+1	
6368	021312	104400	.WORD .S.	
6369	021314		MOVE MEM,BREG	;RESTORE BREG...
6370		000022	MICPC=MICPC+1	
6371	021314	040620	.WORD .S.	
6372	021316		SBR 25	;LOOP ON ERROR...
6373		000023	MICPC=MICPC+1	
6374	021316	100403	.WORD .S.	
6375	021320		CALL SCP1	;IS LOOP DATA SET???
6376	021320		MOVE # <MICPC+3>,BREG	
6377		000024	MICPC=MICPC+1	
6378	021320	000426	.WORD .S.	
6379	021322		SBR SCP1	
6380		000025	MICPC=MICPC+1	
6381	021322	104427	.WORD .S.	
6382	021324		MOVE MEM,BREG	;RESTORE BREG...
6383		000026	MICPC=MICPC+1	
6384	021324	040620	.WORD .S.	
6385	021326		SBR 25	;LOOP ON DATA.
6386		000027	MICPC=MICPC+1	
6387	021326	100403	.WORD .S.	

35:

```

6388 021330          MOVE    MEM,BREG          ;RESTORE BREG...
6389          MICPC=MICPC+1
6390 021330 000030      .WORD    .S.
6391 021332          SHFBRT          ;SET THE NEXT BIT...
6392          MICPC=MICPC+1
6393 021332 000031      .WORD    SBR!.SELB!.DBRSH
6394 021334          BBT    45          ;BRANCH IF DONE...
6395          MICPC=MICPC+1
6396 021334 000032      .WORD    .S.
6397 021336          SBR    25          ;CONTINUE...
6398          MICPC=MICPC+1
6399 021336 000033      .WORD    .S.
6400 021340          45:
6401 021340          SSPFLT 0,115,125,135,145,11
6402 021340          115: MOVE    #177,BREG ;START WITH BIT 7.
6403          MICPC=MICPC+1
6404 021340 000034      .WORD    .S.
6405 021342 000577      125: MOVE    BREG,SPAD <11> ;LOAD THE SCRATCH PAD...
6406          MICPC=MICPC+1
6407 021342 000035      .WORD    .S.
6408 021344 063231      MOVE    SPAD <11>,MEM ;GET THE "FOUND"...
6409          MICPC=MICPC+1
6410 021344 000036      .WORD    .S.
6411 021346          MOVE    MEM,SPAD <0> ;
6412          MICPC=MICPC+1
6413 021346 000037      .WORD    .S.
6414 021350 043220      MOVE    BREG,MEM          ;SAVE THE CONTENTS OF BREG...
6415          MICPC=MICPC+1
6416 021350 000040      .WORD    .S.
6417 021352 062620      $IFEQ  BREG,SPAD <0> 135 ;IF GOOD.. CONTINUE...
6418          MICPC=MICPC+1
6419          SUB2C  SPAD <0>,BREG,NOP
6420 021352          MICPC=MICPC+1
6421          .WORD    .S.
6422 021352 000041      BZ    135
6423 021354 060360      MICPC=MICPC+1
6424          .WORD    .S.
6425 021354 000042      MOVE    BREG,OUT1 <CSR4> ;ELSE, REPORT ERROR...
6426 021356 101456      MICPC=MICPC+1
6427          .WORD    .S.
6428 021356 000043      MOVE    SPAD <0>,BREG ;
6429 021360 061224      MICPC=MICPC+1
6430          .WORD    .S.
6431 021360 000044      MOVE    BREG,OUT1 <CSR5> ;BAD DATA...
6432 021362 060600      MICPC=MICPC+1
6433          .WORD    .S.
6434 021362 000045      MOVE    #4,BREG          ;TYPE OF ERROR...
6435 021364 061225      MICPC=MICPC+1
6436 021364 000046      MOVE    BREG,OUT1 <CSR3> ;
6437 021366 000404      MICPC=MICPC+1
6438          .WORD    .S.
6439 021366 000047      MOVE    #11,BREG          ;LOAD ADDRESS.
6440 021370 061223      MICPC=MICPC+1
6441          .WORD    .S.
6442 021370 000050
6443 021370 000411

```

6444 021372
6445 021372
6446 021372
6447 021374
6448 021374
6449 021374
6450 021374
6451 021376
6452 021376
6453 021400
6454 021400
6455 021400
6456 021402
6457 021402
6458 021404
6459 021404
6460 021404
6461 021404
6462 021406
6463 021406
6464 021406
6465 021410
6466 021410
6467 021410
6468 021412
6469 021412
6470 021412
6471 021412
6472 021412
6473 021414
6474 021414
6475 021414
6476 021416
6477 021416
6478 021420
6479 021420
6480 021420
6481 021420
6482 021420
6483 021420
6484 021420
6485 021420
6486 021420
6487 021420
6488 021420
6489 021420
6490 021420
6491 021420
6492 021420
6493 021420
6494 021420
6495 021420
6496 021420
6497 021420
6498 021420
6499 021420
6500 021420

000051
061227
000052
000454
000053
104400
000054
040620
000055
100435
000056
000460
000057
104427
000060
040620
000061
100435
000062
040620
000063
061620
000064
103435
000065
000467
000066
104454
000067
100400

```

MOVE BREG OUT1 <CSR7> ;
NICPC=NICPC+1
.WORD .S.
CALL EROR ;DATA ERROR!!
MOVE B <NICPC+3>,BREG
NICPC=NICPC+1
.WORD .S.
SBR EROR
NICPC=NICPC+1
.WORD .S.
MOVE MEM,BREG ;RESTORE BREG...
NICPC=NICPC+1
.WORD .S.
SBR 125 ;LOOP ON ERROR...
NICPC=NICPC+1
.WORD .S.
CALL SCP1 ;IS LOOP DATA SET??
MOVE B <NICPC+3>,BREG
NICPC=NICPC+1
.WORD .S.
SBR SCP1
NICPC=NICPC+1
.WORD .S.
MOVE MEM,BREG ;RESTORE BREG...
NICPC=NICPC+1
.WORD .S.
SBR 125 ;LOOP ON DATA.
NICPC=NICPC+1
.WORD .S.
MOVE MEM,BREG ;RESTORE BREG...
NICPC=NICPC+1
.WORD .S.
SHFBT ;SET THE NEXT BIT...
NICPC=NICPC+1
.WORD SBR! .SELB! .DBRSH
BB7 125 ;BRANCH IF NOT DONE...
NICPC=NICPC+1
.WORD .S.
145: CALL SCPE
MOVE B <NICPC+3>,BREG
NICPC=NICPC+1
.WORD .S.
SBR SCPE
NICPC=NICPC+1
.WORD .S.
SBR 215
NICPC=NICPC+1
.WORD .S.
SSPTS1 12
SXZ

```

***** TEST 26 *****
;# SCRATCH PAD TEST FOR SP12
;# FLOAT A 1 THROUGH SCRATCH PAD 12

KMC11 SCRATCH PAD TESTS

```

6500 ;* FLOAT A 0 THROUGH SCRATCH PAD 12
6501 021430 SXZ
6502 ;:*****
6503
6504 021430 STSTN
6505 ; TEST 26
6506
6507 021430 012737 000026 001202 TST26: MOV #26,STSTNM ; LOAD THE NO. OF THIS TEST
6508 021436 012737 021644 001442 MOV #STST27,NEXT ; POINT TO THE START OF NEXT TEST.
6509 ;R1 CONTAINS BASE KMC11 ADDRESS
6510 021444 004737 035536 JSR PC,LDVINT ;LOAD-VERIFY-WAIT.
6511 021450 021464 MCT26
6512 021452 104022 ERROR 22 ;TIME OUT ERROR...
6513 021454 012706 001200 MOV #STACK,SP ;RESET STACK...
6514 021460 000177 157756 JMF #NEXT ;GO TO NEXT TEST...
6515
6516
6517 021464 MOVE #0,BREG ;SET TO CLEAR SPAD 16...
6518 000000 NICPC=NICPC+1
6519 021464 000400 .WORD $
6520 021466 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES.
6521 000001 NICPC=NICPC+1
6522 021466 063236 .WORD $
6523
6524 021470 SSPFLT 1,15,25,35,45,12
6525 021470 15: MOVE #200,BREG ;START WITH BIT 7.
6526 000002 NICPC=NICPC+1
6527 021472 000600 .WORD $
6528 25: MOVE BREG,SPAD <12> ;LOAD THE SCRATCH PAD...
6529 000003 NICPC=NICPC+1
6530 021472 063232 .WORD $
6531 021474 MOVE SPAD <12>,MEM ;GET THE "FOUND"...
6532 000004 NICPC=NICPC+1
6533 021474 062612 .WORD $
6534 021476 MOVE MEM,SPAD <0> ;
6535 000005 NICPC=NICPC+1
6536 021476 043220 .WORD $
6537 021478 MOVE BREG,MEM ;SAVE THE CONTENTS OF BREG...
6538 000006 NICPC=NICPC+1
6539 021500 062620 .WORD $
6540 021502 $IFEQ BREG,SPAD <0> 35 ;IF GOOD.. CONTINUE...
6541
6542
6543 021502 SUB2C SPAD <0>,BREG,NOP
6544 000007 NICPC=NICPC+1
6545 021502 060360 .WORD $
6546 021504 BZ 35
6547 000010 NICPC=NICPC+1
6548 021504 101424 .WORD $
6549 021506 MOVE BREG,OUT1 <CSR4> ;ELSE, REPORT ERROR...
6550 000011 NICPC=NICPC+1
6551 021506 061224 .WORD $
6552 021510 MOVE SPAD <0>,BREG ;
6553 000012 NICPC=NICPC+1
6554 021512 060600 .WORD $
6555 000013 MOVE BREG,OUT1 <CSR5> ;BAD DATA...
6556 NICPC=NICPC+1

```

6556 021512 061225
 6557 021514
 6558 021514 000014
 6559 021514 000404
 6560 021516 000015
 6561 021516 061223
 6562 021520 000016
 6563 021520 000412
 6564 021522 000017
 6565 021522 061227
 6566 021524
 6567 021524 000020
 6568 021524 000422
 6569 021526 000021
 6570 021526 104400
 6571 021528 000022
 6572 021528 040620
 6573 021530 000023
 6574 021530 100403
 6575 021532 000024
 6576 021532 000426
 6577 021534 000025
 6578 021534 104427
 6579 021536 000026
 6580 021536 040620
 6581 021540 000027
 6582 021540 100403
 6583 021542 000030
 6584 021542 040620
 6585 021544 000031
 6586 021544 061620
 6587 021546 000032
 6588 021546 103434
 6589 021552 000033
 6590 021552 100403
 6591 021554
 6592 021554
 6593 021554
 6594 021554 000034
 6595 021554 000577

```

.WORD .S.
MOVE # 4, BREG ;TYPE OF ERROR...
NICPC=NICPC+1
.WORD .S.
MOVE BREG, OUT1 (CSR3) ;
NICPC=NICPC+1
.WORD .S.
MOVE # 12, BREG ;LOAD ADDRESS.
NICPC=NICPC+1
.WORD .S.
MOVE BREG, OUT1 (CSR7) ;
CALL EROR ;DATA ERROR!!
MOVE # (NICPC+3), BREG
NICPC=NICPC+1
.WORD .S.
SER EROR
MOVE NEH, BREG ;RESTORE BREG...
NICPC=NICPC+1
.WORD .S.
SER 25 ;LOOP ON ENROR...
NICPC=NICPC+1
.WORD .S.
CALL SCP1 ;IS LOOP DATA SET???
MOVE # (NICPC+3), BREG
NICPC=NICPC+1
.WORD .S.
SER SCP1
NICPC=NICPC+1
.WORD .S.
MOVE NEH, BREG ;RESTORE BREG...
NICPC=NICPC+1
.WORD .S.
SER 25 ;LOOP ON DATA.
MOVE NEH, BREG ;RESTORE BREG...
NICPC=NICPC+1
.WORD .S.
SHFRT ;SET THE NEXT BIT...
NICPC=NICPC+1
.WORD .SBR! .SELB! .DBRSH
BB7 45 ;BRANCH IF DONE...
NICPC=NICPC+1
.WORD .S.
SER 25 ;CONTINUE...
NICPC=NICPC+1
.WORD .S.
45:
SSPFLT 0, 115, 125, 135, 145, 12
115:
MOVE # 177, BREG ;START WITH BIT 7.
NICPC=NICPC+1
.WORD .S.

```

KMC11 SCRATCH PAD TESTS

6612	021556		125:	MOVE	BREG	SPAD <12>	;LOAD THE SCRATCH PAD...
6613		000035		MICPC=	MICPC+1		
6614	021556	063232		.WORD	.S.		
6615	021560			MOVE	SPAD <12>	,MEM	;GET THE "FOUND"...
6616		000036		MICPC=	MICPC+1		
6617	021560	062612		.WORD	.S.		
6618	021562			MOVE	MEM	SPAD <0>	;
6619		000037		MICPC=	MICPC+1		
6620	021562	043220		.WORD	.S.		
6621	021564			MOVE	BREG	MEM	;SAVE THE CONTENTS OF BREG...
6622		000040		MICPC=	MICPC+1		
6623	021564	062620		.WORD	.S.		
6624	021566			SIFEQ	BREG	,SPAD <0>	135 ;IF GOOD.. CONTINUE...
6625							
6626							
6627	021566			SUBC	SPAD <0>	,BREG,NOP	
6628		000041		MICPC=	MICPC+1		
6629	021566	060360		.WORD	.S.		
6630	021570			BZ	135		
6631		000042		MICPC=	MICPC+1		
6632	021570	101456		.WORD	.S.		
6633	021572			MOVE	BREG	OUT1 <CSR4>	:ELSE, REPORT ERROR...
6634		000043		MICPC=	MICPC+1		
6635	021572	061224		.WORD	.S.		
6636	021574			MOVE	SPAD <0>	,BREG	;
6637		000044		MICPC=	MICPC+1		
6638	021574	060600		.WORD	.S.		
6639	021576			MOVE	BREG	OUT1 <CSRS>	;BAD DATA...
6640		000045		MICPC=	MICPC+1		
6641	021576	061225		.WORD	.S.		
6642	021600			MOVE	# 4	BREG	;TYPE OF ERROR...
6643		000046		MICPC=	MICPC+1		
6644	021600	000404		.WORD	.S.		
6645	021602			MOVE	BREG	OUT1 <CSR3>	;
6646		000047		MICPC=	MICPC+1		
6647	021602	061223		.WORD	.S.		
6648	021604			MOVE	# 12	BREG	;LOAD ADDRESS.
6649		000050		MICPC=	MICPC+1		
6650	021604	000412		.WORD	.S.		
6651	021606			MOVE	BREG	OUT1 <CSR7>	;
6652		000051		MICPC=	MICPC+1		
6653	021606	061227		.WORD	.S.		
6654	021610			CALL	EROR		;DATA ERROR!!
6655	021610			MOVE	#	(MICPC+3),BREG	
6656		000052		MICPC=	MICPC+1		
6657	021610	000454		.WORD	.S.		
6658	021612			SBR	EROR		
6659		000053		MICPC=	MICPC+1		
6660	021612	104400		.WORD	.S.		
6661	021614			MOVE	MEM	BREG	;RESTORE BREG...
6662		000054		MICPC=	MICPC+1		
6663	021614	040620		.WORD	.S.		
6664	021616			SBR	125		;LOOP ON ERROR...
6665		000055		MICPC=	MICPC+1		
6666	021616	100435		.WORD	.S.		
6667	021620		135:	CALL	SCP1		;IS LOOP DATA SET???

KMC11 SCRATCH PAD TESTS

```

6668 021620 MOVE      # <MICPC+3>,BREG
6669          MICPC=MICPC+1
6670 021620 .WORD      .S.
6671 021622 SBR       SCP1
6672          MICPC=MICPC+1
6673 021622 .WORD      .S.
6674 021624 MOVE      MEM,BREG          ;RESTORE BREG...
6675          MICPC=MICPC+1
6676 021624 .WORD      .S.
6677 021626 SBR       12$          ;LOOP ON DATA.
6678          MICPC=MICPC+1
6679 021626 .WORD      .S.
6680 021630 MOVE      MEM,BREG          ;RESTORE BREG...
6681          MICPC=MICPC+1
6682 021630 .WORD      .S.
6683 021632 SHFBT          ;SET THE NEXT BIT...
6684          MICPC=MICPC+1
6685 021632 .WORD      SBR!..SELB!..DBRSH
6686 021634 BBT       12$          ;BRANCH IF NOT DONE...
6687          MICPC=MICPC+1
6688 021634 .WORD      .S.
6689          14$:
6690 021636 CALL      SCPE
6691 021636 MOVE      # <MICPC+3>,BREG
6692          MICPC=MICPC+1
6693 021636 .WORD      .S.
6694 021640 SBR       SCPE
6695          MICPC=MICPC+1
6696 021640 .WORD      .S.
6697 021642 SBR       21$
6698          MICPC=MICPC+1
6699 021642 .WORD      .S.
6700 021644 SSPTS1  13
6701 021644 SXZ
6702
6703
6704
6705          ;***** TEST 27 *****
6706          ;* SCRATCH PAD TEST FOR SP13
6707          ;* FLOAT A 1 THROUGH SCRATCH PAD 13
6708          ;* FLOAT A 0 THROUGH SCRATCH PAD 13
6709          ;*****
6710
6711          STSTN
6712          ; TEST 27
6713          ;-----
6714 021644 012737 000027 001202 TST27: MOV      #27,STSTNM          ; LOAD THE NO. OF THIS TEST
6715 021652 012737 022060 001442 MOV      #TST30,NEXT          ; POINT TO THE START OF NEXT TEST.
6716          ;R1 CONTAINS BASE KMC11 ADDRESS
6717 021660 004737 035536 JSR      PC,LDVRMT          ;LOAD-VERIFY-WAIT.
6718 021664 021700 MCT27
6719 021666 104022 ERROR    22          ;TIME OUT ERROR...
6720 021670 012706 001200 MOV      #STACK,SP          ;RESET STACK
6721 021674 000177 157542 JMP      @NEXT          ;GO TO NEXT TEST...
6722 021700
6723 021700 MCT27:
          21$:

```


6724	021700		MOVE # 0, BREG ;SET TO CLEAR SPAD 16...
6725		000000	MICPC=MICPC+1
6726	021700	000400	.WORD .S.
6727	021702		MOVE BREG, SPAD <16> ;FOR RETURN ADDRESS PURPOSES.
6728		000001	MICPC=MICPC+1
6729	021702	063236	.WORD .S.
6730	021704		SSPFLT 1, 15, 25, 35, 45, 13
6731	021704		15: MOVE # 200, BREG ;START WITH BIT 7.
6732		000002	MICPC=MICPC+1
6733	021704	000600	.WORD .S.
6734	021706		25: MOVE BREG, SPAD <13> ;LOAD THE SCRATCH PAD...
6735		000003	MICPC=MICPC+1
6736	021706	063233	.WORD .S.
6737	021710		MOVE SPAD <13>, MEM ;GET THE "FOUND"...
6738		000004	MICPC=MICPC+1
6739	021710	062613	.WORD .S.
6740	021712		MOVE MEM, SPAD <0> ;
6741		000005	MICPC=MICPC+1
6742	021712	043220	.WORD .S.
6743	021714		MOVE BREG, MEM ;SAVE THE CONTENTS OF BREG...
6744		000006	MICPC=MICPC+1
6745	021714	062620	.WORD .S.
6746	021716		\$IFEQ BREG, SPAD <0> 3\$;IF GOOD.. CONTINUE...
6747			
6748			
6749	021716		SUB2C SPAD <0>, BREG, NOP
6750		000007	MICPC=MICPC+1
6751	021716	060360	.WORD .S.
6752	021720		BZ 3\$
6753		000010	MICPC=MICPC+1
6754	021720	101424	.WORD .S.
6755	021722		MOVE BREG, OUT1 <CSR4> ;ELSE, REPORT ERROR...
6756		000011	MICPC=MICPC+1
6757	021722	061224	.WORD .S.
6758	021724		MOVE SPAD <0>, BREG ;
6759		000012	MICPC=MICPC+1
6760	021724	060600	.WORD .S.
6761	021726		MOVE BREG, OUT1 <CSR5> ;BAD DATA...
6762		000013	MICPC=MICPC+1
6763		061225	.WORD .S.
6764		000014	MOVE # 4, BREG ;TYPE OF ERROR...
6765		060404	MICPC=MICPC+1
6766			.WORD .S.
6767			MOVE BREG, OUT1 <CSR3> ;
6768		000015	MICPC=MICPC+1
6769		061223	.WORD .S.
6770			MOVE # 13, BREG ;LOAD ADDRESS.
6771		000016	MICPC=MICPC+1
6772	021734	060413	.WORD .S.
6773	021736		MOVE BREG, OUT1 <CSR7> ;
6774		000017	MICPC=MICPC+1
6775	021736	061227	.WORD .S.
6776	021740		CALL EROR ;DATA ERROR!!
6777	021740		MOVE # <MICPC+3>, BREG
6778		000020	MICPC=MICPC+1
6779	021740	000422	.WORD .S.

KMC11 SCRATCH PAD TESTS

6780	021742		SBR	EROR	
6781		000021	MICPC=MICPC+1		
6782	021742	104400	.WORD	.S.	
6783	021744		MOVE	MEM,BREG	;RESTORE BREG...
6784		000022	MICPC=MICPC+1		
6785	021744	040620	.WORD	.S.	
6786	021746		SBR	25	;LOOP ON ERROR...
6787		000023	MICPC=MICPC+1		
6788	021746	100403	.WORD	.S.	
6789	021750		35: CALL	SCP1	;IS LOOP DATA SET???
6790	021750		MOVE	8<MICPC+3>,BREG	
6791		000024	MICPC=MICPC+1		
6792	021750	000426	.WORD	.S.	
6793	021752		SBR	SCP1	
6794		000025	MICPC=MICPC+1		
6795	021752	104427	.WORD	.S.	
6796	021754		MOVE	MEM,BREG	;RESTORE BREG...
6797		000026	MICPC=MICPC+1		
6798	021754	040620	.WORD	.S.	
6799	021756		SBR	25	;LOOP ON DATA.
6800		000027	MICPC=MICPC+1		
6801	021756	100403	.WORD	.S.	
6802	021760		MOVE	MEM,BREG	;RESTORE BREG...
6803		000030	MICPC=MICPC+1		
6804	021760	040620	.WORD	.S.	
6805	021762		SFBRT		;SET THE NEXT BIT...
6806		000031	MICPC=MICPC+1		
6807	021762	061620	.WORD	SBR!.SELB!.DBRSH	
6808	021764		SBR	45	;BRANCH IF DONE...
6809		000032	MICPC=MICPC+1		
6810	021764	103434	.WORD	.S.	
6811	021766		SBR	25	;CONTINUE...
6812		000033	MICPC=MICPC+1		
6813	021766	100403	.WORD	.S.	
6814	021770		45: SSPFLT	0,115,125,135,145,13	
6815	021770		115: MOVE	8,177,BREG	;START WITH BIT 7.
6816	021770		MICPC=MICPC+1		
6817		000034	.WORD	.S.	
6818	021770	000577	125: MOVE	BREG,SPAD<13>	;LOAD THE SCRATCH PAD...
6819	021772		MICPC=MICPC+1		
6820		000035	.WORD	.S.	
6821	021772	063233	MOVE	SPAD<13>,MEM	;GET THE "FOUND"...
6822	021774		MICPC=MICPC+1		
6823		000036	.WORD	.S.	
6824	021774	062613	MOVE	MEM,SPAD<0>	;
6825	021776		MICPC=MICPC+1		
6826		000037	.WORD	.S.	
6827	021776	043220	MOVE	BREG,MEM	;SAVE THE CONTENTS OF BREG...
6828	022000		MICPC=MICPC+1		
6829		000040	.WORD	.S.	
6830	022000	062620	SIFEQ	BREG,SPAD<0> 135	;IF GOOD.. CONTINUE...
6831	022002				
6832					
6833					
6834	022002		SUB2C	SPAD<0>,BREG,NOP	
6835		000041	MICPC=MICPC+1		

6836	022002	060360	.WORD .S.	
6837	022004		MOV #13	
6838		000042	MICPC=MICPC+1	
6839	022004	101456	.WORD .S.	
6840	022006		MOVE BREG,OUT1 <CSR4>	;ELSE, REPORT ERROR...
6841		000043	MICPC=MICPC+1	
6842	022006	061224	.WORD .S.	
6843	022010		MOVE SPAD <0>,BREG	;
6844		000044	MICPC=MICPC+1	
6845	022010	060600	.WORD .S.	
6846	022012		MOVE BREG,OUT1 <CSR5>	;BAD DATA...
6847		000045	MICPC=MICPC+1	
6848	022012	061225	.WORD .S.	
6849	022014		MOVE #4,BREG	;TYPE OF ERROR...
6850		000046	MICPC=MICPC+1	
6851	022014	000404	.WORD .S.	
6852	022016		MOVE BREG,OUT1 <CSR3>	;
6853		000047	MICPC=MICPC+1	
6854	022016	061223	.WORD .S.	
6855	022020		MOVE #13,BREG	;LOAD ADDRESS.
6856		000050	MICPC=MICPC+1	
6857	022020	000413	.WORD .S.	
6858	022022		MOVE BREG,OUT1 <CSR7>	;
6859		000051	MICPC=MICPC+1	
6860	022022	061227	.WORD .S.	
6861	022024		CALL EROR	;DATA ERROR!!
6862	022024		MOVE # <MICPC+3>,BREG	
6863		000052	MICPC=MICPC+1	
6864	022024	000454	.WORD .S.	
6865	022026		SBR EROR	
6866		000053	MICPC=MICPC+1	
6867	022026	104400	.WORD .S.	
6868	022030		MOVE MEM,BREG	;RESTORE BREG...
6869		000054	MICPC=MICPC+1	
6870	022030	040620	.WORD .S.	
6871	022032		SBR 125	;LOOP ON ERROR...
6872		000055	MICPC=MICPC+1	
6873	022032	100435	.WORD .S.	
6874	022034		CALL SCP1	;IS LOOP DATA SET???
6875	022034		MOVE # <MICPC+3>,BREG	
6876		000056	MICPC=MICPC+1	
6877	022034	000460	.WORD .S.	
6878	022036		SBR SCP1	
6879		000057	MICPC=MICPC+1	
6880	022036	104427	.WORD .S.	
6881	022040		MOVE MEM,BREG	;RESTORE BREG...
6882		000060	MICPC=MICPC+1	
6883	022040	040620	.WORD .S.	
6884	022042		SBR 125	;LOOP ON DATA.
6885		000061	MICPC=MICPC+1	
6886	022042	100435	.WORD .S.	
6887	022044		MOVE MEM,BREG	;RESTORE BREG...
6888		000062	MICPC=MICPC+1	
6889	022044	040620	.WORD .S.	
6890	022046		SHFBRT	;SET THE NEXT BIT...
6891		000063	MICPC=MICPC+1	

135:

KMC11 SCRATCH PAD TESTS

```

6892 022046 061620 .WORD .SBR!.SELB!.DBRSH
6893 022050 BB7 125 ;BRANCH IF NOT DONE...
6894 000064 MICPC=MICPC+1
6895 022050 103435 .WORD .S.
6896 022052 145: CALL SCPE
6897 022052 MOVE # <MICPC+3>,BREG
6898 022052 MICPC=MICPC+1
6899 000065 .WORD .S.
6900 022052 000467 .SBR SCPE
6901 022054 MICPC=MICPC+1
6902 000066 .WORD .S.
6903 022054 104454 .SBR 215
6904 022056 MICPC=MICPC+1
6905 000067 .WORD .S.
6906 022056 100400 SSPTS1 14
6907 022060 SXZ
6908 022060
6909
6910
6911
6912 ;***** TEST 30 *****
6913 ;* SCRATCH PAD TEST FOR SP14
6914 ;* FLOAT A 1 THROUGH SCRATCH PAD 14
6915 ;* FLOAT A 0 THROUGH SCRATCH PAD 14
6916 022060 SXZ
6917 ;*****
6918 022060 STSTN
6919 ; TEST 30
6920
6921 022060 012737 000030 001202 TST30: MOV #30,STSTNM ; LOAD THE NO. OF THIS TEST
6922 022066 012737 022274 001442 MOV #TST31,NEXT ; POINT TO THE START OF NEXT TEST.
6923
6924 022074 004737 035536 JSR PC,LDVRNT ;R1 CONTAINS BASE KMC11 ADDRESS
6925 022100 022114 MCT30 ;LOAD-VERIFY-WAIT.
6926 022102 104022 ERROR 22 ;TIME OUT ERROR...
6927 022104 012706 001200 MOV #STACK,SP ;RESET STACK...
6928 022110 000177 157326 JMP #NEXT ;GO TO NEXT TEST...
6929 022114 MCT30:
6930 022114 215:
6931 022114 MOVE # 0,BREG ;SET TO CLEAR SPAD 16...
6932 000000 MICPC=MICPC+1
6933 022114 000400 .WORD .S.
6934 022116 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES.
6935 000001 MICPC=MICPC+1
6936 022116 063236 .WORD .S.
6937 022120 SSPFLT 1,15,25,35,45,14
6938 022120 15: MOVE # 200,BREG ;START WITH BIT 7.
6939 000002 MICPC=MICPC+1
6940 022120 000600 .WORD .S.
6941 022122 25: MOVE BREG,SPAD <14> ;LOAD THE SCRATCH PAD...
6942 000003 MICPC=MICPC+1
6943 022122 063234 .WORD .S.
6944 022124 MOVE SPAD <14>,MEM ;GET THE "FOUND"...
6945 000004 MICPC=MICPC+1
6946 022124 062614 .WORD .S.
6947 022126 MOVE MEM,SPAD <0> ;

```

6948		000005	MICPC=MICPC+1	
6949	022126	043220	.WORD .S.	
6950	022130		MOVE BREG, MEM	;SAVE THE CONTENTS OF BREG...
6951		000006	MICPC=MICPC+1	
6952	022130	062620	.WORD .S.	
6953	022132		SIFEQ BREG, SPAD <0> 3S	;IF GOOD.. CONTINUE...
6954				
6955				
6956	022132		SUB2C SPAD <0>, BREG, NOP	
6957		000007	MICPC=MICPC+1	
6958	022132	060360	.WORD .S.	
6959	022134		BZ 3S	
6960		000010	MICPC=MICPC+1	
6961	022134	101424	.WORD .S.	
6962	022136		MOVE BREG, OUT1 <CSR4>	;ELSE, REPORT ERROR...
6963		000011	MICPC=MICPC+1	
6964	022136	061224	.WORD .S.	
6965	022140		MOVE SPAD <0>, BREG	;
6966		000012	MICPC=MICPC+1	
6967	022140	060600	.WORD .S.	
6968	022142		MOVE BREG, OUT1 <CSR5>	;BAD DATA...
6969		000013	MICPC=MICPC+1	
6970	022142	061225	.WORD .S.	
6971	022144		MOVE # 4, BREG	;TYPE OF ERROR...
6972		000014	MICPC=MICPC+1	
6973	022144	000404	.WORD .S.	
6974	022146		MOVE BREG, OUT1 <CSR3>	;
6975		000015	MICPC=MICPC+1	
6976	022146	061223	.WORD .S.	
6977	022150		MOVE # 14, BREG	;LOAD ADDRESS.
6978		000016	MICPC=MICPC+1	
6979	022150	000414	.WORD .S.	
6980	022152		MOVE BREG, OUT1 <CSR7>	;
6981		000017	MICPC=MICPC+1	
6982	022152	061227	.WORD .S.	
6983	022154		CALL EROR	;DATA ERROR!!
6984	022154		MOVE # <MICPC+3>, BREG	
6985		000020	MICPC=MICPC+1	
6986	022154	000422	.WORD .S.	
6987	022156		SBR EROR	
6988		000021	MICPC=MICPC+1	
6989	022156	104400	.WORD .S.	
6990	022160		MOVE MEM, BREG	;RESTORE BREG...
6991		000022	MICPC=MICPC+1	
6992	022160	040620	.WORD .S.	
6993	022162		SBR 2S	;LOOP ON ERROR...
6994		000023	MICPC=MICPC+1	
6995	022162	100403	.WORD .S.	
6996	022164		CALL SCP1	;IS LOOP DATA SET???
6997	022164		MOVE # <MICPC+3>, BREG	
6998		000024	MICPC=MICPC+1	
6999	022164	000426	.WORD .S.	
7000	022166		SBR SCP1	
7001		000025	MICPC=MICPC+1	
7002	022166	104427	.WORD .S.	
7003	022170		MOVE MEM, BREG	;RESTORE BREG...

KNC11 SCRATCH PAD TESTS

7004		000026	NICPC=NICPC+1	
7005	022170	040620	.WORD .S.	
7006	022172		SBR 25	; LOOP ON DATA.
7007		000027	NICPC=NICPC+1	
7008	022172	100403	.WORD .S.	
7009	022174		MOVE MEM,BREG	; RESTORE BREG...
7010		000030	NICPC=NICPC+1	
7011	022174	040620	.WORD .S.	
7012	022176		SHFBT	; SET THE NEXT BIT...
7013		000031	NICPC=NICPC+1	
7014	022176	061620	.WORD SBR!..SELB!..DBRSH	
7015	022200		B87 45	; BRANCH IF DONE...
7016		000032	NICPC=NICPC+1	
7017	022200	103434	.WORD .S.	
7018	022202		SBR 25	; CONTINUE...
7019		000033	NICPC=NICPC+1	
7020	022202	100403	.WORD .S.	
7021	022204			
7022	022204		45: SSPFLT 0,118,125,135,145,14	
7023	022204		118: MOVE # 177,BREG ; START WITH BIT 7.	
7024		000034	NICPC=NICPC+1	
7025	022204	000577	.WORD .S.	
7026	022206		125: MOVE BREG,SPAD <14>	; LOAD THE SCRATCH PAD...
7027		000035	NICPC=NICPC+1	
7028	022206	063234	.WORD .S.	
7029	022210		MOVE SPAD <14>,MEM	; GET THE "FOUND"...
7030		000036	NICPC=NICPC+1	
7031	022210	062614	.WORD .S.	
7032	022212		MOVE MEM,SPAD <0>	;
7033		000037	NICPC=NICPC+1	
7034	022212	043220	.WORD .S.	
7035	022214		MOVE BREG, MEM	; SAVE THE CONTENTS OF BREG...
7036		000040	NICPC=NICPC+1	
7037	022214	062620	.WORD .S.	
7038	022216		SIFE0 BREG,SPAD <0> 135	; IF GOOD.. CONTINUE...
7039				
7040				
7041	022216		SUB2C SPAD <0>,BREG,NOP	
7042		000041	NICPC=NICPC+1	
7043	022216	060360	.WORD .S.	
7044	022220		BZ 135	
7045		000042	NICPC=NICPC+1	
7046	022220	101456	.WORD .S.	
7047	022222		MOVE BREG,OUT1 <CSR4>	; ELSE, REPORT ERROR...
7048		000043	NICPC=NICPC+1	
7049	022222	061224	.WORD .S.	
7050	022224		MOVE SPAD <0>,BREG	;
7051		000044	NICPC=NICPC+1	
7052	022224	060600	.WORD .S.	
7053	022226		MOVE BREG,OUT1 <CSR5>	; BAD DATA...
7054		000045	NICPC=NICPC+1	
7055	022226	061225	.WORD .S.	
7056	022230		MOVE # 4,BREG	; TYPE OF ERROR...
7057		000046	NICPC=NICPC+1	
7058	022230	000404	.WORD .S.	
7059	022232		MOVE BREG,OUT1 <CSR3>	;

7060		000047	MICPC=MICPC+1	
7061	022232	061223	.WORD .S	
7062	022234		MOVE #14,BREG	;LOAD ADDRESS.
7063		000050	MICPC=MICPC+1	
7064	022234	000414	.WORD .S	
7065	022236		MOVE BREG,OUT1 (CSR7)	;
7066		000051	MICPC=MICPC+1	
7067	022236	061227	.WORD .S	
7068	022240		CALL EROR	;DATA ERROR!!
7069	022240		MOVE # (MICPC+3),BREG	
7070		000052	MICPC=MICPC+1	
7071	022240	000454	.WORD .S	
7072	022242		SBR EROR	
7073		000053	MICPC=MICPC+1	
7074	022242	104400	.WORD .S	
7075	022244		MOVE MEM,BREG	;RESTORE BREG...
7076		000054	MICPC=MICPC+1	
7077	022244	040620	.WORD .S	
7078	022246		SBR 12\$;LOOP ON ERROR...
7079		000055	MICPC=MICPC+1	
7080	022246	100435	.WORD .S	
7081	022250		CALL SCP1	;IS LOOP DATA SET???
7082	022250		MOVE # (MICPC+3),BREG	
7083		000056	MICPC=MICPC+1	
7084	022250	000460	.WORD .S	
7085	022252		SBR SCP1	
7086		000057	MICPC=MICPC+1	
7087	022252	104427	.WORD .S	
7088	022254		MOVE MEM,BREG	;RESTORE BREG...
7089		000060	MICPC=MICPC+1	
7090	022254	040620	.WORD .S	
7091	022256		SBR 12\$;LOOP ON DATA.
7092		000061	MICPC=MICPC+1	
7093	022256	100435	.WORD .S	
7094	022260		MOVE MEM,BREG	;RESTORE BREG...
7095		000062	MICPC=MICPC+1	
7096	022260	040620	.WORD .S	
7097	022262		SBR 12\$;SET THE NEXT BIT...
7098		000063	MICPC=MICPC+1	
7099	022262	061620	.WORD .SBR!..SELB!..DBRSH	
7100	022264		SBR 12\$;BRANCH IF NOT DONE...
7101		000064	MICPC=MICPC+1	
7102	022264	103435	.WORD .S	
7103	022266			
7104	022266		CALL SCPE	
7105	022266		MOVE # (MICPC+3),BREG	
7106		000065	MICPC=MICPC+1	
7107	022266	000467	.WORD .S	
7108	022270		SBR SCPE	
7109		000066	MICPC=MICPC+1	
7110	022270	104454	.WORD .S	
7111	022272		SBR 21\$	
7112		000067	MICPC=MICPC+1	
7113	022272	100400	.WORD .S	
7114	022274			
7115	022274			

SSPTS1 15
SXZ

```

7116
7117
7118
7119
7120
7121
7122 022274          SXZ
7123
7124
7125 022274          STSTN
7126
7127
7128 022274 012737 000031 001202 TST31:
7129 022302 012737 022510 001442      MOV      #31,STSTNM          ; LOAD THE NO. OF THIS TEST
7130
7131 022310 004737 035536          JSR      PC,LDRWT          ; POINT TO THE START OF NEXT TEST.
7132 022314 022330          MCT31
7133 022316 104022          ERROR    22                ; R1 CONTAINS BASE KMC11 ADDRESS
7134 022320 012706 001200          MOV      #STACK,SP        ; LOAD-VERIFY-WAIT.
7135 022324 000177 157112          JMP      @NEXT            ; TIME OUT ERROR...
7136
7137 022330          MCT31:
7138 022330          215:
7139
7140 022330 000000          MOVE     #0,BREG          ; SET TO CLEAR SPAD 16...
7141 022332 000400          MICPC=MICPC+1
7142
7143 022332 000001          .WORD   $
7144 022334 063236          MOVE     BREG,SPAD <16> ; FOR RETURN ADDRESS PURPOSES.
7145 022334          MICPC=MICPC+1
7146
7147 022334 000002          .WORD   $
7148 022336 000600          SSPFLT  1,15,25,35,45,15 ; START WITH BIT 7.
7149
7150 022336 000003          15:
7151 022340 063235          MOVE     #200,BREG
7152
7153 022340 000004          MICPC=MICPC+1
7154 022342 062615          .WORD   $
7155
7156 022342 000005          MOVE     SPAD <15>,MEM    ; LOAD THE SCRATCH PAD...
7157 022344 043220          MICPC=MICPC+1
7158
7159 022344 000006          .WORD   $
7160 022346 062620          MOVE     MEM,SPAD <0>    ; GET THE "FOUND"...
7161
7162
7163 022346          25:
7164 022346 000007          MOVE     BREG,MEM        ; SAVE THE CONTENTS OF BREG...
7165 022346 060360          MICPC=MICPC+1
7166 022350          .WORD   $
7167 022350 000010          BZ      35
7168 022350 101424          MICPC=MICPC+1
7169 022352          .WORD   $
7170 022352 000011          MOVE     BREG,OUT1 <CSR4> ; ELSE, REPORT ERROR...
7171 022352 061224          MICPC=MICPC+1
7172
7173
7174
7175
7176
7177
7178
7179
7180
7181
7182
7183
7184
7185
7186
7187
7188
7189
7190
7191
7192
7193
7194
7195
7196
7197
7198
7199
7200

```


7172	022354		MOVE SPAD <0>,BREG	;
7173		000012	NICPC=NICPC+1	
7174	022354	060600	.WORD .S.	
7175	022356		MOVE BREG,OUT1 <CSRS>	;BAD DATA...
7176		000013	NICPC=NICPC+1	
7177	022356	061225	.WORD .S.	
7178	022360		MOVE 8 4,BREG	;TYPE OF ERROR...
7179		000014	NICPC=NICPC+1	
7180	022360	000404	.WORD .S.	
7181	022362		MOVE BREG,OUT1 <CSR3>	;
7182		000015	NICPC=NICPC+1	
7183	022362	061223	.WORD .S.	
7184	022364		MOVE 8 15,BREG	;LOAD ADDRESS.
7185		000016	NICPC=NICPC+1	
7186	022364	000415	.WORD .S.	
7187	022366		MOVE BREG,OUT1 <CSR7>	;
7188		000017	NICPC=NICPC+1	
7189	022366	061227	.WORD .S.	
7190	022370		CALL ENDR	;DATA ERROR!!
7191	022370		MOVE 8 <NICPC+3>,BREG	
7192		000020	NICPC=NICPC+1	
7193	022370	000422	.WORD .S.	
7194	022372		CALL ENDR	
7195		000021	NICPC=NICPC+1	
7196	022372	104400	.WORD .S.	
7197	022374		MOVE NEH,BREG	;RESTORE BREG...
7198		000022	NICPC=NICPC+1	
7199	022374	040620	.WORD .S.	
7200	022376		CALL ENDR	;LOOP ON ERROR...
7201		000023	NICPC=NICPC+1	
7202	022376	100403	.WORD .S.	
7203	022400		CALL SCP1	;IS LOOP DATA SET???
7204	022400		MOVE 8 <NICPC+3>,BREG	
7205		000024	NICPC=NICPC+1	
7206	022400	000425	.WORD .S.	
7207	022402		CALL SCP1	
7208		000025	NICPC=NICPC+1	
7209	022402	104427	.WORD .S.	
7210	022404		MOVE NEH,BREG	;RESTORE BREG...
7211		000026	NICPC=NICPC+1	
7212	022404	040620	.WORD .S.	
7213	022406		CALL ENDR	;LOOP ON DATA.
7214		000027	NICPC=NICPC+1	
7215	022406	100403	.WORD .S.	
7216	022410		MOVE NEH,BREG	;RESTORE BREG...
7217		000030	NICPC=NICPC+1	
7218	022410	040620	.WORD .S.	
7219	022412		SHFBT	;SET THE NEXT BIT...
7220		000031	NICPC=NICPC+1	
7221	022412	061620	.WORD .SBR!..SELB!..DARSH	
7222	022414		BB7 48	;BRANCH IF DONE...
7223		000032	NICPC=NICPC+1	
7224	022414	103434	.WORD .S.	
7225	022416		CALL ENDR	;CONTINUE...
7226		000033	NICPC=NICPC+1	
7227	022416	100403	.WORD .S.	

35:

KMC11 SCRATCH PAD TESTS

7228	022420		45:	
7229	022420		SSPFLT	0 115 125 135 145 15
7230	022420		115:	MOVE # 177 BREG ;START WITH BIT 7.
7231		000034		NICPC=NICPC+1
7232		000577		.WORD .S.
7233	022420		125:	MOVE BREG,SPAD <15> ;LOAD THE SCRATCH PAD...
7234	022422	000035		NICPC=NICPC+1
7235		063235		.WORD .S.
7236	022422			MOVE SPAD <15>,MEM ;GET THE "FOUND"...
7237		000036		NICPC=NICPC+1
7238	022424	062615		.WORD .S.
7239	022426			MOVE MEM,SPAD <0> ;
7240		000037		NICPC=NICPC+1
7241	022426	043220		.WORD .S.
7242	022430			MOVE BREG,MEM ;SAVE THE CONTENTS OF BREG...
7243		000040		NICPC=NICPC+1
7244	022430	062620		.WORD .S.
7245	022432			SIFEQ BREG,SPAD <0> 135 ;IF GOOD.. CONTINUE...
7246				
7247				
7248	022432			SUBC SPAD <0>,BREG,NOP
7249		000041		NICPC=NICPC+1
7250	022432	060360		.WORD .S.
7251	022434			BZ 135
7252		000042		NICPC=NICPC+1
7253	022434	101456		.WORD .S.
7254	022436			MOVE BREG,OUT1 <CSR4> ;ELSE, REPORT ERROR...
7255		000043		NICPC=NICPC+1
7256	022436	061224		.WORD .S.
7257	022440			MOVE SPAD <0>,BREG ;
7258		000044		NICPC=NICPC+1
7259	022440	060600		.WORD .S.
7260	022442			MOVE BREG,OUT1 <CSR5> ;BAD DATA...
7261		000045		NICPC=NICPC+1
7262	022442	061225		.WORD .S.
7263	022444			MOVE # 4,BREG ;TYPE OF ERROR...
7264		000046		NICPC=NICPC+1
7265	022444	000404		.WORD .S.
7266	022446			MOVE BREG,OUT1 <CSR3> ;
7267		000047		NICPC=NICPC+1
7268	022446	061223		.WORD .S.
7269	022450			MOVE # 15,BREG ;LOAD ADDRESS.
7270		000050		NICPC=NICPC+1
7271	022450	000415		.WORD .S.
7272	022452			MOVE BREG,OUT1 <CSR7> ;
7273		000051		NICPC=NICPC+1
7274	022452	061227		.WORD .S.
7275	022454			CALL EROR ;DATA ERROR!!
7276	022454			MOVE # <NICPC+3>,BREG
7277		000052		NICPC=NICPC+1
7278	022454	000454		.WORD .S.
7279	022456			SBR EROR
7280		000053		NICPC=NICPC+1
7281	022456	104400		.WORD .S.
7282	022460			MOVE MEM,BREG ;RESTORE BREG...
7283		000054		NICPC=NICPC+1

7284 022460 040620
7285 022462
7286 000055
7287 022462 100435
7288 022464
7289 022464
7290 000056
7291 022464 000460
7292 022466
7293 000057
7294 022466 104427
7295 022470
7296 000060
7297 022470 040620
7298 022472
7299 000061
7300 022472 100435
7301 022474
7302 000062
7303 022474 040620
7304 022476
7305 000063
7306 022476 061620
7307 022500
7308 000064
7309 022500 103435
7310 022502
7311 022502
7312 022502
7313 000065
7314 022502 000467
7315 022504
7316 000066
7317 022504 104454
7318 022506
7319 000067
7320 022506 100400
7321 022510
7322 022510
7323
7324
7325
7326
7327
7328 022510
7329
7330
7331 022510
7332
7333
7334 022510 012737 000032 001202
7335 022516 012737 022704 001442
7336
7337 022524 004737 035536
7338 022530 022544
7339 022532 104022

```

      .WORD      .S.
SBR      128
      ;LOOP ON ERROR...
      MICPC=MICPC+1
      .WORD      .S.
135:    CALL      SCPI
      MOVE      8 <MICPC+3>,BREG
      MICPC=MICPC+1
      .WORD      .S.
SBR      SCPI
      MICPC=MICPC+1
      .WORD      .S.
      MOVE      MEM,BREG
      MICPC=MICPC+1
      ;RESTORE BREG...
      .WORD      .S.
SBR      128
      MICPC=MICPC+1
      ;LOOP ON DATA.
      .WORD      .S.
      MOVE      MEM,BREG
      MICPC=MICPC+1
      ;RESTORE BREG...
      .WORD      .S.
      SHFBRT
      MICPC=MICPC+1
      ;SET THE NEXT BIT...
      .WORD      .SBR!..SELB!..DBRSH
      BB7      128
      MICPC=MICPC+1
      ;BRANCH IF NOT DCNE...
      .WORD      .S.
145:    CALL      SCPE
      MOVE      8 <MICPC+3>,BREG
      MICPC=MICPC+1
      .WORD      .S.
SBR      SCPE
      MICPC=MICPC+1
      .WORD      .S.
SBR      216
      MICPC=MICPC+1
      .WORD      .S.
SNPR1
SXZ
;***** TEST 32 *****
;* NPR TEST
;* TEST OF DATI, 1 WORD FROM 11 MEMORY TO UPROC
SXZ
;*****
STSTN
; TEST 32
-----
TST32: NOV      #32,STSTNM
      NOV      #ST33,NEXT
      ; LOAD THE NO. OF THIS TEST
      ; POINT TO THE START OF NEXT TEST.
      ;R1 CONTAINS BASE KMC11 ADDRESS
      JSR      PC,LDRHT
      MCT32
      ;LOAD-VERIFY-WAIT.
      ERROR    22
      ;TIME OUT ERROR...

```

7340	022534	012706	001200
7341	022540	000177	156676
7342	022544		
7343	022544		
7344		000000	
7345	022544	000400	
7346	022546		
7347		000001	
7348	022546	063236	
7349	022550		
7350		000002	
7351	022550	062220	
7352	022552		
7353		000003	
7354	022552	062221	
7355	022554		
7356		000004	
7357	022554	062224	
7358	022556		
7359		000005	
7360	022556	062225	
7361	022560		
7362		000006	
7363	022560	061230	
7364	022562		
7365		000007	
7366	022562	000525	
7367	022564		
7368		000010	
7369	022564	063223	
7370	022566		
7371		000011	
7372	022566	061620	
7373	022570		
7374		000012	
7375	022570	063224	
7376	022572		
7377		000013	
7378	022572	000702	
7379	022574		
7380		000014	
7381	022574	063221	
7382	022576		
7383		000015	
7384	022576	000445	
7385	022600		
7386		000016	
7387	022600	063222	
7388	022602		
7389		000017	
7390	022602	060601	
7391	022604		
7392		000020	
7393	022604	062224	
7394	022606		
7395		000021	

NCT32:
18:

```

MOV      @STACK, SP      ; RESET STACK...
JMP      @NEXT          ; GO TO NEXT TEST...

MOVE     @ 0, BREG      ; SET TO CLEAR NPR IBUS REGISTERS.
NICPC=NICPC+1

.MWORD  .S.
MOVE     BREG, SPAD <16> ; FOR RETURN ADDRESS PURPOSE...
NICPC=NICPC+1

.MWORD  .S.
MOVE     BREG, OUT0 <0>  ; CLEAR IBUS+IN DATA+LB.
NICPC=NICPC+1

.MWORD  .S.
MOVE     BREG, OUT0 <1>  ; CLEAR IBUS+IN DATA+HB.
NICPC=NICPC+1

.MWORD  .S.
MOVE     BREG, OUT0 <4>  ; CLEAR INBA+LB.
NICPC=NICPC+1

.MWORD  .S.
MOVE     BREG, OUT0 <5>  ; CLEAR INBA+HB.
NICPC=NICPC+1

.MWORD  .S.
MOVE     BREG, OUT1 <CSR10> ; CLEAR EA+BITS.
NICPC=NICPC+1

.MWORD  .S.
MOVE     @ 125, BREG     ; SET GOOD DATA.
NICPC=NICPC+1

.MWORD  .S.
MOVE     BREG, SPAD <3> ; " " "
NICPC=NICPC+1

SHFBRT
.MWORD  .S.
MOVE     BREG, SPAD <4> ; " " "
NICPC=NICPC+1

.MWORD  .S.
MOVE     @ <NPDAT&377>, BREG ; SET ADDRESS.
NICPC=NICPC+1

.MWORD  .S.
MOVE     BREG, SPAD <1> ; " " "
NICPC=NICPC+1

.MWORD  .S.
MOVE     @ <NPDAT&177400/400>, BREG ;
NICPC=NICPC+1

.MWORD  .S.
MOVE     BREG, SPAD <2> ;
NICPC=NICPC+1

.MWORD  .S.
MOVE     SPAD <1>, BREG ; SET INBA+ADDRESS
NICPC=NICPC+1

.MWORD  .S.
MOVE     BREG, OUT0 <4> ;
NICPC=NICPC+1

.MWORD  .S.
MOVE     SPAD <2>, BREG ; SET INBA+ADDRESS
NICPC=NICPC+1

```

7396	022606	060602	.WORD .S.	
7397	022610		MOVE BREG,OUTD <5>	;
7398		000022	NICPC=NICPC+1	
7399	022610	062225	.WORD .S.	
7400	022612		MOVE B,001,BREG	;SET NPR BIT.
7401		000023	NICPC=NICPC+1	
7402	022612	000401	.WORD .S.	
7403	022614		MOVE BREG,OUT1 <CSR10>	;SET NPR RQ BIT WITH IN NPR & WRD.XFR
7404		000024	NICPC=NICPC+1	
7405	022614	061230	.WORD .S.	
7406	022616		35: MOVE INP1 <CSR10>,BREG	;IS NPR DONE???
7407		000025	NICPC=NICPC+1	
7408	022616	120600	.WORD .S.	
7409	022620		BZ 35	;NO, SPIN ON IT??
7410		000026	NICPC=NICPC+1	
7411	022620	102025	.WORD .S.	
7412	022622		MOVE INPD <0>,BREG	;GET THE DATA LB.
7413		000027	NICPC=NICPC+	
7414	022622	020400	.WORD .S.	
7415	022624		\$IFEQ BREG,SPAD <3> 45	;IF GOOD, CHECK NEXT BYTE.
7416				
7417				
7418	022624		SUB2C SPAD <3>,BREG,NOP	
7419		000030	NICPC=NICPC+1	
7420	022624	060363	.WORD .S.	
7421	022626		BZ 45	
7422		000031	NICPC=NICPC+1	
7423	022626	101442	.WORD .S.	
7424	022630		MOVE BREG,OUT1 <CSR5>	;MOVE GOOD DATA...
7425		000032	NICPC=NICPC+1	
7426	022630	061225	.WORD .S.	
7427	022632		MOVE SPAD <3>,BREG	;MOVE BAD DATA...
7428		000033	NICPC=NICPC+1	
7429	022632	060603	.WORD .S.	
7430	022634		MOVE BREG,OUT1 <CSR4>	;
7431		000034	NICPC=NICPC+1	
7432	022634	061224	.WORD .S.	
7433	022636		MOVE B,11,BREG	;SET THE TYPE OF ERROR...
7434		000035	NICPC=NICPC+1	
7435	022636	000411	.WORD .S.	
7436	022640		MOVE BREG,OUT1 <CSR3>	;
7437		000036	NICPC=NICPC+1	
7438	022640	061223	.WORD .S.	
7439	022642		CALL ERROR1	;REPORT ERROR...
7440	022642		MOVE B <NICPC+3>,BREG	
7441		000037	NICPC=NICPC+1	
7442	022642	000441	.WORD .S.	
7443	022644		SBR ERROR1	
7444		000040	NICPC=NICPC+1	
7445	022644	104401	.WORD .S.	
7446	022646		SBR 15	;
7447		000041	NICPC=NICPC+1	
7448	022646	100400	.WORD .S.	
7449	022650		45: MOVE INPD <1>,BREG	;GET THE NPR DATA MB.
7450		000042	NICPC=NICPC+1	
7451	022650	020420	.WORD .S.	

```

7452 022652 SIFEQ BREG,SPAD <4> SS ;IF GOOD, BRANCH.
7453
7454 022652 SUBPC SPAD <4>,BREG,NOP
7455 000043 MICPC=MICPC+1
7456 060364 .WORD .S.
7457 022652 BZ SS
7458 022654 MICPC=MICPC+1
7459 000044 .WORD .S.
7460 022654 101454 MOVE BREG,OUT1 <CSR5> ;GOOD DATA.
7461 022656 MICPC=MICPC+1
7462 000045 .WORD .S.
7463 022656 061225 MOVE SPAD <4>,OUT1 <CSR4> ;BAD DATA.
7464 022660 MICPC=MICPC+1
7465 000046 .WORD .S.
7466 022660 061204 MOVE # 11, MEM ;TYPE OF ERROR...
7467 022662 MICPC=MICPC+1
7468 000047 .WORD .S.
7469 022662 002411 MOVE MEM,OUT1 <CSR3> ;
7470 022664 MICPC=MICPC+1
7471 000050 .WORD .S.
7472 022664 041223 CALL ERROR1 ;DATA ERROR...
7473 022666 MOVE # <MICPC+3>,BREG
7474 022666 MICPC=MICPC+1
7475 000051 .WORD .S.
7476 022666 000453 SBR ERROR1
7477 022670 MICPC=MICPC+1
7478 000052 .WORD .S.
7479 022670 104401 SBR IS ;LOOP ON ERROR...
7480 022672 MICPC=MICPC+1
7481 000053 .WORD .S.
7482 022672 100400 CALL SCOPE ;SCOPE THE TEST...
7483 022674 SS: MOVE # <MICPC+3>,BREG
7484 022674 MICPC=MICPC+1
7485 000054 .WORD .S.
7486 022674 000456 SBR SCOPE
7487 022676 MICPC=MICPC+1
7488 000055 .WORD .S.
7489 022676 104454 SBR IS ;DO THE NEXT ITERATION...
7490 022700 MICPC=MICPC+1
7491 000056 .WORD .S.
7492 022700 100400 NPOAT: .BYTE 125,252
7493 022702 125 252 ;
7494 022704 SXZ 21,WORD,1
7495 022704 SXZ
7496
7497
7498 ;***** TEST 33 *****
7499 ;* NPR TEST
7500 ;* TEST OF DATO, 1 WORD FROM UPROC TO PDP11 MEMORY.
7501 ;* THEN DATI OF THAT WORD AND CHECK....
7502 022704 SXZ ;*****
7503
7504
7505 022704 STSTN ;
7506 ; TEST 33
7507 ;-----

```

7508	022704	012737	000033	001202	TST33:	MOV	#33,STSTNM				
7509	022712	012737	023154	001442		MOV	#TST34,NEXT				
7510											
7511	022720	004737	035536			JSR	PC,LVWRM				
7512	022724	022740				MCT33					
7513	022726	104022				ERROR	22				
7514	022730	012706	001200			MOV	#STACK,SP				
7515	022734	000177	156502			JMP	#NEXT				
7516	022740				MCT33:						
7517	022740				IS:	MOVE	#0,BREG				
7518		000000				MICPC=MICPC+1					
7519	022740	000400				.WORD	.S.				
7520	022742					MOVE	BREG,SPAD <16>				
7521		000001				MICPC=MICPC+1					
7522	022742	063236				.WORD	.S.				
7523	022744					MOVE	BREG,OUT0 <0>				
7524		000002				MICPC=MICPC+1					
7525	022744	062220				.WORD	.S.				
7526	022746					MOVE	BREG,OUT0 <1>				
7527		000003				MICPC=MICPC+1					
7528	022746	062221				.WORD	.S.				
7529	022750					MOVE	BREG,OUT0 <2>				
7530		000004				MICPC=MICPC+1					
7531	022750	062222				.WORD	.S.				
7532	022752					MOVE	BREG,OUT0 <3>				
7533		000005				MICPC=MICPC+1					
7534	022752	062223				.WORD	.S.				
7535	022754					MOVE	BREG,OUT0 <4>				
7536		000006				MICPC=MICPC+1					
7537	022754	062224				.WORD	.S.				
7538	022756					MOVE	BREG,OUT0 <5>				
7539		000007				MICPC=MICPC+1					
7540	022756	062225				.WORD	.S.				
7541	022760					MOVE	BREG,OUT0 <6>				
7542		000010				MICPC=MICPC+1					
7543	022760	062226				.WORD	.S.				
7544	022762					MOVE	BREG,OUT0 <7>				
7545		000011				MICPC=MICPC+1					
7546	022762	062227				.WORD	.S.				
7547	022764					MOVE	BREG,OUT1 <CSR10>				
7548		000012				MICPC=MICPC+1					
7549	022764	061230				.WORD	.S.				
7550	022766					MOVE	INP1 <CSR11>,SPAD <0>				
7551		000013				MICPC=MICPC+1					
7552	022766	123220				.WORD	.S.				
7553	022770					OR	SPAD <0>,BREG,OUT1 <CSR11>				
7554		000014				MICPC=MICPC+1					
7555	022770	061311				.WORD	.S.				
7556	022772					MOVE	#125,BREG				
7557		000015				MICPC=MICPC+1					
7558	022772	000525				.WORD	.S.				
7559	022774					MOVE	BREG,SPAD <3>				
7560		000016				MICPC=MICPC+1					
7561	022774	063223				.WORD	.S.				
7562	022776					SHFBRT					
7563		000017				MICPC=MICPC+1					

```

; LOAD THE NO. OF THIS TEST
; POINT TO THE START OF NEXT TEST.
; R1 CONTAINS BASE KMC11 ADDRESS
; LOAD-VERIFY-WAIT.
; TIME OUT ERROR...
; RESET STACK...
; GO TO NEXT TEST...

```

```

; SET TO CLEAR NPR IBUS+REGISTERS.

```

```

; F0 RETURN ADDRESS PURPOSE...

```

```

; CLEAR IBUS+IN DATA+LB.

```

```

; CLEAR IBUS+IN DATA+HB.

```

```

; CLEAR IBUS+OUT DATA+LB.

```

```

; CLEAR IBUS+OUT DATA+HB.

```

```

; CLEAR IBUS+IN BA+LB.

```

```

; CLEAR IBUS+IN BA+HB.

```

```

; CLEAR IBUS+OUT BA+LB.

```

```

; CLEAR IBUS+OUT BA+HB.

```

```

; CLEAR IN+BA EABITS.

```

```

; CLEAR OUT+BA EA+BITS.

```

```

; CLEAR OUT+BA EA+BITS.

```

```

; GOOD DATA.

```

7564	022776	061620	.WORD .SBR!.SELB!.DBRSH	
7565	023000		MOVE BREG,SPAD (4)	;
7566		000020	MICPC=MICPC+1	
7567	023000	063224	.WORD .S.	
7568	023002		MOVE # (NPRDTI&377),BREG	;SET NPR ADDRESS.
7569		000021	MICPC=MICPC+1	
7570	023002	000552	.WORD .S.	
7571	023004		MOVE BREG,SPAD (1)	;
7572		000022	MICPC=MICPC+1	
7573	023004	063221	.WORD .S.	
7574	023006		MOVE # (NPRDTI&177400/400),BREG	;
7575		000023	MICPC=MICPC+1	
7576	023006	000446	.WORD .S.	
7577	023010		MOVE BREG,SPAD (2)	;
7578		000024	MICPC=MICPC+1	
7579	023010	063222	.WORD .S.	
7580	023012		MOVE SPAD (1),BREG	;SET OUT+BA+ADDRESS.
7581		000025	MICPC=MICPC+1	
7582	023012	060601	.WORD .S.	
7583	023014		MOVE BREG,OUTO (6)	; " " " "
7584		000026	MICPC=MICPC+1	
7585	023014	062226	.WORD .S.	
7586	023016		MOVE SPAD (2),BREG	;SET OUT+BA+ADDRESS.
7587		000027	MICPC=MICPC+1	
7588	023016	060602	.WORD .S.	
7589	023020		MOVE BREG,OUTO (7)	; " " " "
7590		000030	MICPC=MICPC+1	
7591	023020	062227	.WORD .S.	
7592	023022		MOVE SPAD (3),BREG	;SET OUT+DATA.
7593		000031	MICPC=MICPC+1	
7594	023022	060603	.WORD .S.	
7595	023024		MOVE BREG,OUTO (2)	; " " " "
7596		000032	MICPC=MICPC+1	
7597	023024	062222	.WORD .S.	
7598	023026		MOVE SPAD (4),BREG	; " " " "
7599		000033	MICPC=MICPC+1	
7600	023026	060604	.WORD .S.	
7601	023030		MOVE BREG,OUTO (3)	; " " " "
7602		000034	MICPC=MICPC+1	
7603	023030	062223	.WORD .S.	
7604	023032		MOVE # 021,BREG	;SET NPR+OUT + NPR+RQ.
7605		000035	MICPC=MICPC+1	
7606	023032	000421	.WORD .S.	
7607	023034		MOVE INP1 (CSR10),SPAD (0)	;GET THE CSR10 REGISTER.
7608		000036	MICPC=MICPC+1	
7609	023034	123200	.WORD .S.	
7610	023036		OR BREG,SPAD (0),OUT1 (CSR10);SET NPR+OUT & NPR+RQ & WRD+XFR.	
7611		000037	MICPC=MICPC+1	
7612	023036	061310	.WORD .S.	
7613	023040		MOVE INP1 (CSR10),BREG	;CHECK IF NPR DONE...
7614		000040	MICPC=MICPC+1	
7615	023040	120600	.WORD .S.	
7616	023042		BBO 35	;NO, SPIN ON IT...
7617		000041	MICPC=MICPC+1	
7618	023042	102040	.WORD .S.	
7619			::	

KMC11 NPR TESTS

7620			::	NOW GET IT BACK BY DATI.	
7621			::		
7622	023044			MOVE SPAD (1),BREG	;SET IN+BA ADDRESS.
7623		000042		MICPC=MICPC+1	
7624	023044	060601		.WORD .S.	
7625	023046			MOVE BREG,OUTO (4)	; " " " " .
7626		000043		MICPC=MICPC+1	
7627	023046	062224		.WORD .S.	
7628	023050			MOVE SPAD (2),BREG	; " " " " .
7629		000044		MICPC=MICPC+1	
7630	023050	060602		.WORD .S.	
7631	023052			MOVE BREG,OUTO (5)	; " " " " .
7632		000045		MICPC=MICPC+1	
7633	023052	062225		.WORD .S.	
7634	023054			MOVE # 001,BREG	;SET NPR+RQ.
7635		000046		MICPC=MICPC+1	
7636	023054	000401		.WORD .S.	
7637	023056			MOVE BREG,OUT1 (CSR10)	;SET NPR+IN & NPR+RQ & WRD+XFR.
7638		000047		MICPC=MICPC+1	
7639	023056	061230		.WORD .S.	
7640	023060		4S:	MOVE INP1 (CSR10),BREG	;IS NPR DONE??
7641		000050		MICPC=MICPC+1	
7642	023060	120600		.WORD .S.	
7643	023062			BBO 4S	;NO, SPIN ON IT...
7644		000051		MICPC=MICPC+1	
7645	023062	102050		.WORD .S.	
7646	023064			MOVE INPO (0),BREG	;GET IN DATA+LB.
7647		000052		MICPC=MICPC+1	
7648	023064	020400		.WORD .S.	
7649	023066			MOVE INPO (2),SPAD (0)	;BRET OUT+DATA+LB.
7650		000053		MICPC=MICPC+1	
7651	023066	023040		.WORD .S.	
7652	023070			SIFEQ BREG,SPAD (0) 6S	;BRANCH IF GOOD..
7653					
7654					
7655	023070			SUB2C SPAD (0),BREG,NOP	
7656		000054		MICPC=MICPC+1	
7657	023070	060360		.WORD .S.	
7658	023072			BZ 6S	
7659		000055		MICPC=MICPC+1	
7660	023072	101466		.WORD .S.	
7661	023074			MOVE BREG,OUT1 (CSR4)	;GOOD DATA
7662		000056		MICPC=MICPC+1	
7663	023074	061224		.WORD .S.	
7664	023076			MOVE SPAD (0),BREG	;BAD DATA.
7665		000057		MICPC=MICPC+1	
7666	023076	060600		.WORD .S.	
7667	023100			MOVE BREG,OUT1 (CSR5)	;
7668		000060		MICPC=MICPC+1	
7669	023100	061225		.WORD .S.	
7670	023102			MOVE # 11, MEM	;SET ERROR TYPE...
7671		000061		MICPC=MICPC+1	
7672	023102	002411		.WORD .S.	
7673	023104			MOVE MEM,OUT1 (CSR3)	
7674		000062		MICPC=MICPC+1	
7675	023104	041223		.WORD .S.	

7675 023106
7676 023106
7678 000063
7679 023106 000465
7680 023110
7681 000064
7682 023110 104401
7683 023112
7684 000065
7685 023112 100400
7686 023114
7687 000066
7688 023114 020420
7689 023116
7690 000067
7691 023116 023060
7692 023120
7693
7694
7695 023120
7696 000070
7697 023120 060360
7698 023122
7699 000071
7700 023122 101502
7701 023124
7702 000072
7703 023124 061224
7704 023126
7705 000073
7706 023126 060600
7707 023130
7708 000074
7709 023130 061225
7710 023132
7711 000075
7712 023132 002411
7713 023134
7714 000076
7715 023134 041223
7716 023136
7717 023136
7718 000077
7719 023136 000501
7720 023140
7721 000100
7722 023140 104401
7723 023142
7724 000101
7725 023142 100400
7726 023144
7727 023144
7728 000102
7729 023144 000504
7730 023146
7731 000103

```
CALL ERROR1 ;DATA ERROR...  
MOVE 8 (MICPC+3),BREG  
MICPC=MICPC+1  
.WORD .S.  
SBR ERROR1  
MICPC=MICPC+1  
.WORD .S.  
SBR IS ;LOOP ON ERROR...  
MICPC=MICPC+1  
.WORD .S.  
65: MOVE INPO (1),BREG ;GET IN DATA + HB....  
MICPC=MICPC+1  
.WORD .S.  
MOVE INPO (3),SPAD (0) ;GET OUT+DATA+HB....  
MICPC=MICPC+1  
.WORD .S.  
SIFEQ BREG,SPAD (0) 75 ;BRANCH IF GOOD...  
  
SUBPC SPAD (0),BREG,NOP  
MICPC=MICPC+1  
.WORD .S.  
BZ 75  
MICPC=MICPC+1  
.WORD .S.  
MOVE BREG,OUT1 (CSR4) ;GOOD DATA.  
MICPC=MICPC+1  
.WORD .S.  
MOVE SPAD (0),BREG ;BAD DATA.  
MICPC=MICPC+1  
MOVE BREG,OUT1 (CSR5) ;  
MICPC=MICPC+1  
.WORD .S.  
MOVE 8 11,MEM ;SET ERROR TYPE...  
MICPC=MICPC+1  
.WORD .S.  
MOVE MEM,OUT1 (CSR3) ;  
MICPC=MICPC+1  
.WORD .S.  
CALL ERROR1 ;DATA ERROR...  
MOVE 8 (MICPC+3),BREG  
MICPC=MICPC+1  
.WORD .S.  
SBR ERROR1  
MICPC=MICPC+1  
.WORD .S.  
SBR IS ;LOOP ON ERROR...  
MICPC=MICPC+1  
75: CALL SCPE ;SCOPE THE TEST...  
MOVE 8 (MICPC+3),BREG  
MICPC=MICPC+1  
.WORD .S.  
SBR SCPE  
MICPC=MICPC+1
```

7732	023146	104454				.WORD .S.	
7733	023150					SBR 15	;YES, DO NEXT ITERATION.
7734		000104				MICPC=MICPC+1	
7735	023150	100400				.WORD .S.	
7736	023152	000000				.WORD 0	
7737	023154				NPRDTI:		
7738	023154				SNPR9	21	
7739					SXZ		
7740							
7741							
7742							
7743							
7744							
7745	023154				SXZ		
7746							
7747							
7748	023154				STSTN		
7749							
7750							
7751	023154	012737	000034	001202	TST34:	MOV #34,STSTNM	
7752	023162	012737	023322	001442		MOV #TST35,NEXT	
7753							
7754	023170	004737	035536			JSR PC,LDVWRT	
7755	023174	023210				MCT34	
7756	023176	104022				ERROR 22	
7757	023200	012706	001200			MOV #STACK,SP	
7758	023204	000177	156232			JMP @NEXT	
7759	023210				MCT34:		
7760	023210				15:	MOVE #0,BREG	
7761		000000				MICPC=MICPC+1	
7762	023210	000400				.WORD .S.	
7763	023212					MOVE BREG,SPAD <16>	;FOR RETURN ADDRESS PURPOSES.
7764		000001				MICPC=MICPC+1	
7765	023212	063236				.WORD .S.	
7766	023214					MOVE BREG,SPAD <1>	;CLEAR DELAY COUNT...
7767		000002				MICPC=MICPC+1	
7768	023214	063221				.WORD .S.	
7769	023216					MOVE BREG,OUT0 <0>	
7770		000003				MICPC=MICPC+1	
7771	023216	062220				.WORD .S.	
7772	023220					MOVE BREG,OUT0 <1>	
7773		000004				MICPC=MICPC+1	
7774	023220	062221				.WORD .S.	
7775	023222					MOVE BREG,OUT0 <2>	
7776		000005				MICPC=MICPC+1	
7777	023222	062222				.WORD .S.	
7778	023224					MOVE BREG,OUT0 <3>	
7779		000006				MICPC=MICPC+1	
7780	023224	062223				.WORD .S.	
7781	023226					MOVE BREG,OUT0 <4>	
7782		000007				MICPC=MICPC+1	
7783	023226	062224				.WORD .S.	
7784	023230					MOVE BREG,OUT0 <5>	
7785		000010				MICPC=MICPC+1	
7786	023230	062225				.WORD .S.	
7787	023232					MOVE BREG,OUT0 <6>	

```

***** TEST 34 *****
;NPR NON-EXISTENT MEMORY TEST
;DO A DATO TO A NON-EXISTENT ADDRESS.
;VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
;*****

```

```

; TEST 34
-----
; LOAD THE NO. OF THIS TEST
; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
;LOAD-VERIFY-WAIT.
;TIME OUT ERROR...
;RESET STACK.
;GO TO NEXT TEST...
;SET TO CLEAR IBUS+REGISTERS.

```

7789 023232 062226
 7790 023234
 7791 023234 000012
 7792 023234 062227
 7793 023236
 7794 000013
 7795 061230
 7796 023240
 7797 000014
 7798 000010
 7799 023240
 7800 023241
 7801 023244
 7802 023244
 7803 023244
 7804 023244 000016
 7805 023246 060776
 7806 023246 000017
 7807 023246 062227
 7808 023250
 7809 000020
 7810 023250 002414
 7811 023252
 7812 000021
 7813 023252 123220
 7814 023254
 7815 000022
 7816 023254 041311
 7817 023256
 7818 000023
 7819 023256 000421
 7820 023260
 7821 000024
 7822 023260 061230
 7823 023262
 7824 000025
 7825 023262 120620
 7826 023264
 7827 000026
 7828 023264 102037
 7829 023266
 7830 000027
 7831 023266 063061
 7832 023270
 7833 000030
 7834 023270 101032
 7835 023272
 7836 000031
 7837 023272 100425
 7838 023274
 7839 000032
 7840 023274 002413
 7841 023276
 7842 000033
 7843 023276 041223

```

MICPC=MICPC+1
.WORD .S.
MOVE BREG,OUT0 <7> ;
MICPC=MICPC+1
.WORD .S.
MOVE BREG,OUT1 <CSR10> ;
MICPC=MICPC+1
.WORD .S.
25: MOVE #320,BREG ;NPR ADDRESS+LB.
MICPC=MICPC+1
.WORD .S.
MOVE BREG,OUT0 <6> ;LOAD OUT+BA+LB.
MICPC=MICPC+1
.WORD .S.
MOVE #376,BREG ;NPR ADDRESS+HB.
MICPC=MICPC+1
.WORD .S.
35: MOVE BREG,OUT0 <7> ;LOAD OUT+BA+HB.
MICPC=MICPC+1
.WORD .S.
MOVE #14,MEM ;SET OUT+BA BITS 16 &17.
MICPC=MICPC+1
.WORD .S.
MOVE INP1 <CSR11>,SPAD <0> ;SET OUT+BA 16+17
MICPC=MICPC+1
.WORD .S.
OR MEM,SPAD <0>,OUT1 <CSR11> ;SET OUT+BA 16 &17.
MICPC=MICPC+1
.WORD .S.
MOVE #21,BREG ;
MICPC=MICPC+1
.WORD .S.
MOVE BREG,OUT1 <CSR10> ;SET NPR+RO OUT+NPR,WRD+XFR...
MICPC=MICPC+1
.WORD .S.
35: MOVE INP1 <CSR11>,BREG ;CHECK IF NON+EX+MEM SET...
MICPC=MICPC+1
.WORD .S.
BBO 45 ;YES, ITS WORKING.
MICPC=MICPC+1
.WORD .S.
SINC SPAD <1> ;WAIT FOR ATLEAST 18USEC.
MICPC=MICPC+1
.WORD .S.
BC 65 ;ENOUGH REPORT ERROR...
MICPC=MICPC+1
.WORD .S.
SBR 35 ;CHECK AGAIN...
MICPC=MICPC+1
.WORD .S.
65: MOVE #13,MEM ;SET ERROR TYPE.
MICPC=MICPC+1
.WORD .S.
MOVE MEM,OUT1 <CSR3> ;
MICPC=MICPC+1
.WORD .S.

```

```

7844 023300 CALL ERROR1 ;REPORT ERROR...
7845 023300 MOVE # <MICPC+3>,BREG
7846 000034 MICPC=MICPC+1
7847 023300 .WORD .S.
7848 023302 JSR ERROR1
7849 000035 MICPC=MICPC+1
7850 023302 104401 .WORD .S.
7851 023304 JSR IS ;LOOP ON ERROR.
7852 000036 MICPC=MICPC+1
7853 023304 100400 .WORD .S.
7854 023306 4S: MOVE # 0,BREG ;
7855 000037 MICPC=MICPC+1
7856 023306 000400 .WORD .S.
7857 023310 MOVE INP1 <CSR11>,SPAD <0> ;
7858 000040 MICPC=MICPC+1
7859 023310 123220 .WORD .S.
7860 023312 AND SPAD <0>,BREG,OUT1 <CSR11> ;RESET NON+EX+MEM.
7861 000041 MICPC=MICPC+1
7862 023312 061271 .WORD .S.
7863 023314 CALL SCPE ;SCOPE THE TEST.
7864 023314 MOVE # <MICPC+3>,BREG
7865 000042 MICPC=MICPC+1
7866 023314 000444 .WORD .S.
7867 023316 JSR SCPE
7868 000043 MICPC=MICPC+1
7869 023316 104454 .WORD .S.
7870 023320 JSR IS ;DO NEXT ITERATION.
7871 000044 MICPC=MICPC+1
7872 023320 100400 .WORD .S.
7873 023322 SNPR4
7874 023322 SXZ
7875
7876
7877 ;***** TEST 35 *****
7878 ;#NPR TEST
7879 ;#TEST OF MULTIPLE NPR'S DOING DATI.
7880 ;#XFER 6 WORD'S FROM 11 MEMORY TO UPROC.
7881 ;#
7882 023322 SXZ
7883
7884 ;*****
7885 023322 STSTN
7886
7887 ; TEST 35
7888 023322 012737 000035 001202 TST35: MOV #35,STSTNM ; LOAD THE NO. OF THIS TEST
7889 023330 012737 023606 001442 MOV #TST36,NEXT ; POINT TO THE START OF NEXT TEST.
7890
7891 023336 004737 035536 JSR PC,LDRWT ;R1 CONTAINS BASE KMC11 ADDRESS
7892 023342 023356 MCT35 ;LOAD-VERIFY-WAIT.
7893 023344 104022 ERROR 22 ;TIME OUT ERROR...
7894 023346 012706 001200 MOV #STACK,SP ;RESET STACK
7895 023352 000177 156064 JMP @NEXT ;GO TO NEXT TEST...
7896 023356 MCT35:
7897 023356 1S: MOVE # 5,SPAD <5> ;SET THE COUNT
7898 000000 MICPC=MICPC+1
7899 023356 003005 .WORD .S.

```

7900	023360		MOVE	# 13 SPAD <13>	;
7901		000001	NICPC=NICPC+1		
7902	023360	003013	.WORD	.S.	
7903	023362		MOVE	# 0, BREG	;SET TO CLEAR NPR IBNS+REGISTERS.
7904		000002	NICPC=NICPC+1		
7905	023362	000400	.WORD	.S.	
7906	023364		MOVE	BREG SPAD <16>	;FOR RETURN ADDRESS PURPOSES...
7907		000003	NICPC=NICPC+1		
7908	023364	063236	.WORD	.S.	
7909	023366		MOVE	BREG OUT0 <0>	;CLEAR IBUS
7910		000004	NICPC=NICPC+1		
7911	023366	062220	.WORD	.S.	
7912	023370		MOVE	BREG OUT0 <1>	;CLEAR IBUS.
7913		000005	NICPC=NICPC+1		
7914	023370	062221	.WORD	.S.	
7915	023372		MOVE	BREG OUT0 <4>	;CLEAR IBUS.
7916		000006	NICPC=NICPC+1		
7917	023372	062224	.WORD	.S.	
7918	023374		MOVE	BREG OUT0 <5>	;CLEAR IBUS.
7919		000007	NICPC=NICPC+1		
7920	023374	062225	.WORD	.S.	
7921	023376		MOVE	BREG OUT1 <CSR10>	;CLEAR EA+BITS
7922		000010	NICPC=NICPC+1		
7923	023376	061230	.WORD	.S.	
7924	023400		MOVE	# <INPDT&377>, BREG	;SET NPR ADDRESS.
7925		000011	NICPC=NICPC+1		
7926	023400	000572	.WORD	.S.	
7927	023402		MOVE	BREG SPAD <1>	;
7928		000012	NICPC=NICPC+1		
7929	023402	063221	.WORD	.S.	
7930	023404		MOVE	# <INPDT&177400/400>, BREG	;
7931		000013	NICPC=NICPC+1		
7932	023404	000447	.WORD	.S.	
7933	023406		MOVE	BREG SPAD <2>	;
7934		000014	NICPC=NICPC+1		
7935	023406	063222	.WORD	.S.	
7936	023410		MOVE	# 125, BREG	;GET READY TO LOAD MEMORY.
7937		000015	NICPC=NICPC+1		
7938	023410	000525	.WORD	.S.	
7939	023412		MOVE	# 0, MLR	;
7940		000016	NICPC=NICPC+1		
7941	023412	010000	.WORD	.S.	
7942	023414		MOVE	# 2, NPR	;
7943		000017	NICPC=NICPC+1		
7944	023414	004002	.WORD	.S.	
7945	023416		125: MOVE	BREG MEM, MARINC	;LOAD THE MEMORY...
7946		000020	NICPC=NICPC+1		
7947	023416	076620	.WORD	.S.	
7948	023420		SWFRT		;SET ALTERNATE 125 OR 252.
7949		000021	NICPC=NICPC+1		
7950	023420	061620	.WORD	.SBR!..SELB!..DBRSH	
7951	023422		SDEC	SPAD <13>	;IS IT DONE???
7952		000022	NICPC=NICPC+1		
7953	023422	063173	.WORD	.S.!..DSP	
7954	023424		BZ	158	;YES, START THE TEST.
7955		000023	NICPC=NICPC+1		

7956	023424	101425	.WORD	.S.	
7957	023425		SBR	125	;NO, CONTINUE.
7958	023426	000024	NICPC=NICPC+1		
7959	023427	100420	.WORD	.S.	
7960	023428		15S: MOVE	0,MLR	;
7961	023429	000025	NICPC=NICPC+1		
7962	023430	010000	.WORD	.S.	
7963	023431		MOVE	0,NPR	;
7964	023432	000026	NICPC=NICPC+1		
7965	023433	004000	.WORD	.S.	
7966	023434		2S: MOVE	SPAD (1),BREG	;SET IN BA ADDRESS.
7967	023435	000027	NICPC=NICPC+1		
7968	023436	060601	.WORD	.S.	
7969	023437		MOVE	BREG,OUTO (4)	;SET IN+BA ADDRESS.
7970	023438	000030	NICPC=NICPC+1		
7971	023439	062224	.WORD	.S.	
7972	023440		MOVE	SPAD (2),BREG	;SET IN+BA ADDRESS.
7973	023441	000031	NICPC=NICPC+1		
7974	023442	060602	.WORD	.S.	
7975	023443		MOVE	BREG,OUTO (5)	;SET IN+BA ADDRESS.
7976	023444	000032	NICPC=NICPC+1		
7977	023445	062225	.WORD	.S.	
7978	023446		MOVE	03,BREG	;SET NOT+LAST+XFR & NPR.R1
7979	023447	000033	NICPC=NICPC+1		
7980	023448	000403	.WORD	.S.	
7981	023449		MOVE	BREG,OUT1 (CSR10)	;SET NOT+LAST+XFR & NPR R1 & WRB.XFR.
7982	023450	000034	NICPC=NICPC+1		
7983	023451	061230	.WORD	.S.	
7984	023452		3S: MOVE	INP1 (CSR10),BREG	;IS NPR DONE??
7985	023453	000035	NICPC=NICPC+1		
7986	023454	120600	.WORD	.S.	
7987	023455		BBO	35	;NO,SPIN ON IT
7988	023456	000036	NICPC=NICPC+1		
7989	023457	102035	.WORD	.S.	
7990	023458		MOVE	INPO (0),MEM MARINC	;
7991	023459	000037	NICPC=NICPC+1		
7992	023460	036400	.WORD	.S.	
7993	023461		MOVE	INPO (1),MEM MARINC	;
7994	023462	000040	NICPC=NICPC+1		
7995	023463	036420	.WORD	.S.	
7996	023464		SDEC	SPAD (5)	;IS IT DONE??
7997	023465	000041	NICPC=NICPC+1		
7998	023466	063165	.WORD	.S.!.DSP	
7999	023467		BZ	45	;YES, GO CHECK THE DATA.
8000	023468	000042	NICPC=NICPC+1		
8001	023469	101447	.WORD	.S.	
8002	023470		SINC	SPAD (1)	;UPDATE ADDRESS.
8003	023471	000043	NICPC=NICPC+1		
8004	023472	063061	.WORD	.S.!.DSP	
8005	023473		SINC	SPAD (1)	;
8006	023474	000044	NICPC=NICPC+1		
8007	023475	063061	.WORD	.S.!.DSP	
8008	023476		SADC	SPAD (2)	;UPDATE ADDRESS.
8009	023477	000045	NICPC=NICPC+1		
8010	023478	063102	.WORD	.S.!.DSP	
8011	023479		SBR	25	;DO NEXT NPR.

8012		000046	MICPC=MICPC+1	
8013	023472	100427	.WORD .S.	
8014	023474		45: MOVE # 0,BREG ;MOVE # 5,SPAD (8)	
8015		000047	MICPC=MICPC+1	
8016	023474	000400	.WORD .S.	
8017	023476		MOVE # 13,SPAD (13) ;	
8018		000050	MICPC=MICPC+1	
8019	023476	003013	.WORD .S.	
8020	023500		MOVE BREG,OUT1 (CSR10) ;CLEAR NOT+LAST+XFR & NPR RQ.	
8021		000051	MICPC=MICPC+1	
8022	023500	061230	.WORD .S.	
8023	023502		MOVE BREG,SPAD (3) ;	
8024		000052	MICPC=MICPC+1	
8025	023502	063223	.WORD .S.	
8026	023504		MOVE BREG,SPAD (0)	
8027		000053	MICPC=MICPC+1	
8028	023504	063220	.WORD .S.	
8029	023506		MOVE BREG,SPAD (1)	
8030		000054	MICPC=MICPC+1	
8031	023506	063221	.WORD .S.	
8032	023510		MOVE # 2,SPAD (2) ;	
8033		000055	MICPC=MICPC+1	
8034	023510	003002	.WORD .S.	
8035	023512		MOVE # 0,MLR ;RESET MAR.	
8036		000056	MICPC=MICPC+1	
8037	023512	010000	.WORD .S.	
8038	023514		MOVE # 0,NPR ;RESTE MAR.	
8039		000057	MICPC=MICPC+1	
8040	023514	004000	.WORD .S.	
8041	023516		55: MOVE MEM,SPAD (0) ;GET THE NPR DATA.	
8042		000060	MICPC=MICPC+1	
8043	023516	043220	.WORD .S.	
8044	023520		MOVE SPAD (1),MLR ;SET GOOD DATA ADDRESS.	
8045		000061	MICPC=MICPC+1	
8046	023520	070201	.WORD .S.	
8047	023522		MOVE SPAD (2),NPR ;SET GOOD DATA ADDRESS.	
8048		000062	MICPC=MICPC+1	
8049	023522	064202	.WORD .S.	
8050	023524		SIFEQ MEM,SPAD (0) 6S ;IF GOOD CHECK NEXT...	
8051				
8052				
8053	023524		SUB2C SPAD (0),MEM,NOP	
8054		000063	MICPC=MICPC+1	
8055	023524	040360	.WORD .S.	
8056	023526		BZ 6S	
8057		000064	MICPC=MICPC+1	
8058	023526	101475	.WORD .S.	
8059	023530		MOVE MEM,OUT1 (CSR4) ;GOOD DATA.	
8060		000065	MICPC=MICPC+1	
8061	023530	041224	.WORD .S.	
8062	023532		MOVE SPAD (0),BREG ;BAD DATA.	
8063		000066	MICPC=MICPC+1	
8064	023532	060600	.WORD .S.	
8065	023534		MOVE BREG,OUT1 (CSRS) ;	
8066		000067	MICPC=MICPC+1	
8067	023534	061225	.WORD .S.	


```

8068 023536          MOVE      # 11 MEM          ;ERROR TYPE...
8069          MICPC=MICPC+1
8070 023536 000070   .WORD      .S.
8071 023540 002411   MOVE      MEM,OUT1 (CSR3);
8072          MICPC=MICPC+1
8073 023540 041223   .WORD      .S.
8074 023542          CALL      ERROR1          ;DATA ERROR...
8075 023542          MOVE      # <MICPC+3>,BREG
8076          MICPC=MICPC+1
8077 023542 000474   .WORD      .S.
8078 023544          SBR      ERROR1
8079          MICPC=MICPC+1
8080 023544 104401   .WORD      .S.
8081 023546          SBR      IS
8082          MICPC=MICPC+1          ;LOOP ON ERROR...
8083 023546 100400   .WORD      .S.
8084 023550          SDEC     SPAD (13)          ;IS IT DONE??
8085          MICPC=MICPC+1
8086 023550 063173   .WORD      .S.!.DSP
8087 023552          BZ      7S
8088          MICPC=MICPC+1          ;YES, SCOPE IT.
8089 023552 101503   .WORD      .S.
8090 023554          $INC     SPAD (3),,MLRLD ;NPR DATA ADDRESS.
8091          MICPC=MICPC+1
8092 023554 073063   .WORD      .S.!.DSP!MLRLD
8093 023556          $INC     SPAD (1)          ;UPDATE GOOD DATA ADDRESS.
8094          MICPC=MICPC+1
8095 023556 063061   .WORD      .S.!.DSP
8096 023560          $INC     SPAD (2)          ;UPDATA GOOD DATA ADDRESS.
8097          MICPC=MICPC+1
8098 023560 063102   .WORD      .S.!.DSP
8099 023562          SBR      5S
8100          MICPC=MICPC+1          ;CHECK NEXT BYTE.
8101 023562 100460   .WORD      .S.
8102 023564          CALL     SCPE
8103 023564          MOVE     # <MICPC+3>,BREG ;SCOPE IT.
8104          MICPC=MICPC+1
8105 023564 000103   .WORD      .S.
8106 023566          SBR      SCPE
8107          MICPC=MICPC+1
8108 023566 104454   .WORD      .S.
8109 023570          SBR      IS
8110          MICPC=MICPC+1          ;DO NEXT ITERATION.
8111 023570 100400   .WORD      .S.
8112 023572 125125 125125 MNPDT: .WORD 125125,125125,125125 ;
8113 023600 125125 125125 125125 .WORD 125125,125125,125125 ;
8114          :MNPDT1: .BLKW 5
8115 023606          $MEM10
8116 023606          SXZ
8117
8118
8119          ;***** TEST 36 *****
8120          ;* MAIN MEMORY TEST
8121          ;* FLOAT A 0 THROUGH ALL MAIN MEMORY LOCATION...
8122 023606          SXZ
8123          ;*****

```

8124					STSTN			
8125	023606							
8126								
8127								
8128	023606	012737	000036	001202	TST36:	MOV	#36,STSTNM	; LOAD THE NO. OF THIS TEST
8129	023614	012737	023764	001442		MOV	#TST37,NEXT	; POINT TO THE START OF NEXT TEST.
8130								;R1 CONTAINS BASE KMC11 ADDRESS
8131	023622	004737	035536			JSR	PC,L0VRMT	;LOAD-VERIFY-WAIT.
8132	023626	023642				MCT36		
8133	023630	104022				ERROR	22	; TIME OUT ERROR...
8134	023632	012706	001200			MOV	#STACK,SP	; RESET STACK...
8135	023636	000177	155600			JMP	#NEXT	; GO TO NEXT TEST...
8136	023642				MCT36:			
8137	023643				SMPLT			
8138	023642							
8139		000000						
8140	023642	000400				MOVE	# 0,BREG	; SET TO CLEAR SPAD 16...
8141	023644					MICPC=MICPC+1		
8142		000001				.WORD	.S.	
8143	023644	063236				MOVE	BREG,SPAD <16>	; FOR RETURN ADDRESS PURPOSES...
8144	023646					MICPC=MICPC+1		
8145		000002			11S:	MOVE	# 177,BREG	; START WITH BIT 7.
8146	023646	000577				MICPC=MICPC+1		
8147	023650					.WORD	.S.	
8148		000003				MOVE	# 0,MLR	; LOAD MAR+LO.
8149	023650	010000				MICPC=MICPC+1		
8150	023652					.WORD	.S.	
8151		000004				MOVE	# 0,MPR	; LOAD MAR+HI.
8152	023652	004000				MICPC=MICPC+1		
8153	023654					.WORD	.S.	
8154		000005				MOVE	# 0,MEM	
8155	023654	002400				MICPC=MICPC+1		
8156	023656					.WORD	.S.	
8157		000006				MOVE	MEM,SPAD <1>	; SET THE COUNTER...
8158	023656	043221				MICPC=MICPC+1		
8159	023660					.WORD	.S.	
8160		000007				MOVE	MEM,SPAD <3>	; SET THE COUNTER...
8161	023660	043223				MICPC=MICPC+1		
8162						.WORD	.S.	
8163	023662				12S:	MOVE	BREG,MEM	; SET THE BIT IN MEMORY.
8164		000010				MICPC=MICPC+1		
8165	023662	062620				.WORD	.S.	
8166	023664					MOVE	MEM,SPAD <2>	; PUT THE "FOUND" IN SPAD <2>.
8167		000011				MICPC=MICPC+1		
8168	023664	043222				.WORD	.S.	
8169	023666					MOVE	BREG,SPAD <4>	; SAVE THE BREG.
8170		000012				MICPC=MICPC+1		
8171	023666	063224				.WORD	.S.	
8172	023670					SIFEQ	BREG,SPAD <2> 13S	; IF GOOD CONTINUE.
8173								
8174								
8175	023670					SUB2C	SPAD <2>,BREG,NOP	
8176		000013				MICPC=MICPC+1		
8177	023670	060362				.WORD	.S.	
8178	023672					BZ	13S	
8179		000014				MICPC=MICPC+1		

8180	023672	101427	.WORD .S.	
8181	023674		MOVE BREG OUT1 <CSR4>	;GOOD DATA...
8182		000015	NICPC=NICPC+1	
8183	023674	061224	.WORD .S.	
8184	023676		MOVE MEM OUT1 <CSRS>	;BAD DATA...
8185		000016	NICPC=NICPC+1	
8186	023676	041225	.WORD .S.	
8187	023700		MOVE # 6 BREG	;SET TYPE OF ERROR...
8188		000017	NICPC=NICPC+1	
8189	023700	000406	.WORD .S.	
8190	023702		MOVE BREG OUT1 <CSR3>	;
8191		000020	NICPC=NICPC+1	
8192	023702	061223	.WORD .S.	
8193	023704		MOVE SPAD <1>,BREG	;LOAD ADDRESS.
8194		000021	NICPC=NICPC+1	
8195	023704	060601	.WORD .S.	
8196	023706		MOVE BREG OUT1 <CSR7>	;
8197		000022	NICPC=NICPC+1	
8198	023706	061227	.WORD .S.	
8199	023710		CALL EROR	;REPORT ERROR...
8200	023710		MOVE # <NICPC+3>,BREG	
8201		000023	NICPC=NICPC+1	
8202	023710	000425	.WORD .S.	
8203	023712		SBR EROR	
8204		000024	NICPC=NICPC+1	
8205	023712	104400	.WORD .S.	
8206	023714		MOVE SPAD <4>,BREG	;RESTORE BREGISTER...
8207		000025	NICPC=NICPC+1	
8208	023714	060604	.WORD .S.	
8209	023716		SBR 128	;LOOP ON ERROR...
8210		000026	NICPC=NICPC+1	
8211	023716	100410	.WORD .S.	
8212	023720		CALL SCP1	;LOOP ON DATA SET??
8213	023720		MOVE # <NICPC+3>,BREG	
8214		000027	NICPC=NICPC+1	
8215	023720	000431	.WORD .S.	
8216	023722		SBR SCP1	
8217		000030	NICPC=NICPC+1	
8218	023722	104427	.WORD .S.	
8219	023724		MOVE SPAD <4>,BREG	;YES, FIRST RESTORE BREG...
8220		000031	NICPC=NICPC+1	
8221	023724	060604	.WORD .S.	
8222	023726		SBR 128	;SCOPE THE DATA..
8223		000032	NICPC=NICPC+1	
8224	023726	100410	.WORD .S.	
8225	023730		MOVE SPAD <4>,BREG	;RESTORE BREG...
8226		000033	NICPC=NICPC+1	
8227	023730	060604	.WORD .S.	
8228	023732		SHFBRT	;RIGHT SHIFT BY 1 BIT.
8229		000034	NICPC=NICPC+1	
8230	023732	061620	.WORD .SBR!..SELB!..DBRSH	
8231	023734		128 ;BRANCH IF NOT DONE.	
8232		000035	NICPC=NICPC+1	
8233	023734	103410	.WORD .S.	
8234	023736		SINC SPAD <1>	;INCREMENT ADDRESS.
8235		000036	NICPC=NICPC+1	

135:

887

145:

8236	023736	063061			.WORD	.S.!.DSP		
8237	023740	000037				SAC SPAD (3)		
8238		063103				NICPC=NICPC+1		
8239	023740	000040			.WORD	.S.!.DSP		
8240	023742	000040				MOVE # 4, MEM		;PREPARE TO CHECK DONE...
8241		002404				NICPC=NICPC+1		
8242	023742				.WORD	.S.		
8243	023744				\$IFLO	MEM, SPAD (3) 15\$;IS IT DONE???
8244	023744					SUB2C SPAD (3), MEM, NOP		
8245		000041				NICPC=NICPC+1		
8246	023744	040363			.WORD	.S.		
8247	023746				BC	15\$		
8248		000042				NICPC=NICPC+1		
8249	023746	101046			.WORD	.S.		
8250								
8251	023750					MOVE SPAD (1), MLR		;LOAD MAR+LO.
8252		000043				NICPC=NICPC+1		
8253	023750	070201			.WORD	.S.		
8254	023752					MOVE SPAD (3), MPR		;LOAD MAR+HI.
8255		000044				NICPC=NICPC+1		
8256	023752	064203			.WORD	.S.		
8257	023754				SBR	12\$;DO THE NEXT MEMORY LOCATION.
8258		000045				NICPC=NICPC+1		
8259	023754	100410			.WORD	.S.		
8260	023754					CALL SCPE		;SCOPE THE TEST...
8261	023756				15\$:	MOVE # (NICPC+3), BREG		
8262	023756					NICPC=NICPC+1		
8263	023756	000046			.WORD	.S.		
8264	023756	000450				SBR SCPE		
8265	023760					NICPC=NICPC+1		
8266		000047			.WORD	.S.		
8267	023760	104454			SBR	11\$		
8268	023762					NICPC=NICPC+1		
8269		000050			.WORD	.S.		
8270	023762	100402						
8271	023764							
8272	023764				\$MEM11			
8273	023764				\$XZ			
8274								
8275								
8276								
8277								
8278								
8279	023764				\$XZ			
8280								
8281								
8282	023764				\$TSTN			
8283								
8284								
8285	023764	012737	000037	001202	TST37:	MOV #37, \$TSTNM		; LOAD THE NO. OF THIS TEST
8286	023772	012737	024144	001442		MOV #TST40, NEXT		; POINT TO THE START OF NEXT TEST.
8287								;R1 CONTAINS BASE KNC11 ADDRESS
8288	024000	004737	035536			JSR PC, LDVWRT		;LOAD-VERIFY-WAIT.
8289	024004	024020				MCT37		
8290	024006	104022				ERROR 22		;TIME OUT ERROR...
8291	024010	012706	001200			MOV #STACK, SP		;RESET STACK...

```

;***** TEST 37 *****
; * MAIN MEMORY TEST
; * FLOAT A 1 THROUGH ALL MAIN MEMORY LOCATIONS...
;:*****
    
```

KMC11 MAIN MEMORY TESTS

8298	024014	000177	155422
8299	024020		
8300	024020		
8301	024020		
8302	024020	000000	
8303	024020	000400	
8304	024022		
8305	024022	000001	
8306	024022	063236	
8307	024024		
8308	024024	000002	
8309	024024	000600	
8310	024026		
8311	024026	000003	
8312	024026	010000	
8313	024026		
8314	024026	000004	
8315	024030	004000	
8316	024032		
8317	024032	000005	
8318	024032	002400	
8319	024034		
8320	024034	000006	
8321	024034	043221	
8322	024036		
8323	024036	000007	
8324	024036	043223	
8325	024040		
8326	024040	000010	
8327	024040	062620	
8328	024042		
8329	024042	000011	
8330	024042	043222	
8331	024044		
8332	024044	000012	
8333	024044	063224	
8334	024046		
8335	024046	000013	
8336	024046	060362	
8337	024050		
8338	024050	000014	
8339	024050	101427	
8340	024052		
8341	024052	000015	
8342	024052	061224	
8343	024054		
8344	024054	000016	
8345	024054	041225	
8346	024056		
8347	024056	000017	
8348	024060	000406	

```

JMP ANEXT ;GO TO NEXT TEST...
MCT37:
SMFLT 1,15,25,35,45,55
MOVE #0,BREG ;SET TO CLEAR SPAD 16...
NICPC=NICPC+1
.WORD .S.
MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES...
NICPC=NICPC+1
.WORD .S.
15: MOVE #200,BREG ;START WITH BIT 7.
NICPC=NICPC+1
.WORD .S.
MOVE #0,MLR ;LOAD MAR+LO.
NICPC=NICPC+1
.WORD .S.
MOVE #0,MPR ;LOAD MAR+HI.
NICPC=NICPC+1
.WORD .S.
MOVE #0,MEM
NICPC=NICPC+1
.WORD .S.
MOVE MEM,SPAD <1> ;SET THE COUNTER...
NICPC=NICPC+1
.WORD .S.
MOVE MEM,SPAD <3> ;SET THE COUNTER...
NICPC=NICPC+1
.WORD .S.
25: MOVE BREG,MEM ;SET THE BIT IN MEMORY.
NICPC=NICPC+1
.WORD .S.
MOVE MEM,SPAD <2> ;PUT THE "FOUND" IN SPAD <2>.
NICPC=NICPC+1
.WORD .S.
MOVE BREG,SPAD <4> ;SAVE THE BREG.
NICPC=NICPC+1
.WORD .S.
$IFEQ BREG,SPAD <2> 3$ ;IF GOOD CONTINUE.

SUB2C SPAD <2>,BREG,NOP
NICPC=NICPC+1
.WORD .S.
BZ 3$
NICPC=NICPC+1
.WORD .S.
MOVE BREG,OUT1 <CSR4> ;GOOD DATA...
NICPC=NICPC+1
.WORD .S.
MOVE MEM,OUT1 <CSR5> ;BAD DATA...
NICPC=NICPC+1
.WORD .S.
MOVE #6,BREG ;SET TYPE OF ERROR...
NICPC=NICPC+1
.WORD .S.
MOVE BREG,OUT1 <CSR3> ;

```

8348 000020
8349 024060 061223
8350 024062
8351 000021
8352 024062 060601
8353 024064
8354 000022
8355 024064 061227
8356 024066
8357 024066
8358 000023
8359 024066 000425
8360 024070
8361 000024
8362 024070 104400
8363 024072
8364 000025
8365 024072 060604
8366 024074
8367 000026
8368 024074 100410
8369 024076
8370 024076
8371 000027
8372 024076 000431
8373 024100
8374 000030
8375 024100 104427
8376 024102
8377 000031
8378 024102 060604
8379 024104
8380 000032
8381 024104 100410
8382 024106
8383 000033
8384 024106 060604
8385 024110
8386 000034
8387 024110 061620
8388 024112
8389 000035
8390 024112 103437
8391 024114
8392 000036
8393 024114 100410
8394 024116
8395 000037
8396 024116 063061
8397 024120
8398 000040
8399 024120 063103
8400 024122
8401 000041
8402 024122 002404
8403 024124

```
MICPC=MICPC+1  
WORD .S.  
MOVE SPAD <1>,BREG ;LOAD ADDRESS.  
MICPC=MICPC+1  
WORD .S.  
MOVE BREG,OUT1 <CSR7> ;  
MICPC=MICPC+1  
WORD .S.  
CALL EROR ;REPORT ERROR...  
MOVE # <MICPC+3>,BREG  
MICPC=MICPC+1  
WORD .S.  
SBR EROR  
MICPC=MICPC+1  
WORD .S.  
MOVE SPAD <4>,BREG ;RESTORE BREGISTER...  
MICPC=MICPC+1  
WORD .S.  
SBR 25 ;LOOP ON ERROR...  
MICPC=MICPC+1  
WORD .S.  
CALL SCP1 ;LOOP ON DATA SET??  
MOVE # <MICPC+3>,BREG  
MICPC=MICPC+1  
WORD .S.  
SBR SCP1  
MICPC=MICPC+1  
WORD .S.  
MOVE SPAD <4>,BREG ;YES, FIRST RESTORE BREG...  
MICPC=MICPC+1  
WORD .S.  
SBR 25 ;SCOPE THE DATA..  
MICPC=MICPC+1  
WORD .S.  
MOVE SPAD <4>,BREG ;RESTORE BREG...  
MICPC=MICPC+1  
WORD .S.  
SHFBT ;RIGHT SHIFT BY 1 BIT.  
MICPC=MICPC+1  
WORD .SBR!..SELB!..DBRSH  
BB7 45 ;BRANCH IF DONE...  
MICPC=MICPC+1  
WORD .S.  
SBR 25 ;ELSE, CONTINUE.  
MICPC=MICPC+1  
WORD .S.  
SINC SPAD <1> ;INCREMENT ADDRESS.  
MICPC=MICPC+1  
WORD .S!..DSP ;  
SADC SPAD <3> ;  
MICPC=MICPC+1  
WORD .S!..DSP ;  
MOVE # 4, MEM ;PREPARE TO CHECK DONE...  
MICPC=MICPC+1  
WORD .S.  
SIFLO MEM,SPAD <3> 55 ;IS IT DONE???
```

KMC11 MAIN MEMORY TESTS

Address	Hex	Hex	Hex	Hex	Hex	Code	Comments
8404							
8405							
8406	024124					SUB2C SPAD <3>,MEM,NOP	
8407		000042				MICPC=MICPC+1	
8408	024124	040363				.WORD .S.	
8409	024126					BC 55	
8410		000043				MICPC=MICPC+1	
8411	024126	101047				.WORD .S.	
8412							
8413	024130					MOVE SPAD <1>,MLR	;LOAD MAR+LO.
8414		000044				MICPC=MICPC+1	
8415	024130	070201				.WORD .S.	
8416	024132					MOVE SPAD <3>,MPR	;LOAD MAR+HI.
8417		000045				MICPC=MICPC+1	
8418	024132	064203				.WORD .S.	
8419	024134					SBR 25	;DO THE NEXT MEMORY LOCATION.
8420		000046				MICPC=MICPC+1	
8421	024134	100410				.WORD .S.	
8422	024136					CALL SCPE	;SCOPE THE TEST...
8423	024136					MOVE # <MICPC+3>,BREG	
8424		000047				MICPC=MICPC+1	
8425	024136	000451				.WORD .S.	
8426	024140					SBR SCPE	
8427		000050				MICPC=MICPC+1	
8428	024140	104454				.WORD .S.	
8429	024142					SBR 15	
8430		000051				MICPC=MICPC+1	
8431	024142	100402				.WORD .S.	
8432	024144						
8433	024144						
8434							
8435							
8436							
8437							
8438							
8439							
8440	024144						
8441							
8442							
8443	024144						
8444							
8445							
8446	024144	012737	000040	001202			
8447	024152	012737	024362	001442	TST40:	MOV #40,STSTN	; LOAD THE NO. OF THIS TEST
8448						MOV #TST41,NEXT	; POINT TO THE START OF NEXT TEST.
8449	024160	004737	035536			JSR PC,LDRWT	;R1 CONTAINS BASE KMC11 ADDRESS
8450	024164	024200				MCT40	;LOAD-VERIFY-WAIT.
8451	024166	104022				ERROR 22	
8452	024170	012706	001200			MOV #STACK,SP	;TIME OUT ERROR...
8453	024174	000177	155242			JMP #NEXT	;RESET STACK...
8454	024200						;GO TO NEXT TEST...
8455	024200				MCT40:		
8456		000000			65:	MOVE #0,BREG	;START AT ADDRESS 0.
8457	024200	000400				MICPC=MICPC+1	
8458	024202					.WORD .S.	
8459		000001				MOVE BREG,SPAD <16>	;FOR RETURN ADDRESS PURPOSE...
						MICPC=MICPC+1	

8460	024202	063236	.WORD .S.	
8461	024204		MOVE # 0, MLR	;LOAD MAR+LO.
8462		000002	MICPC=MICPC+1	
8463	024204	010000	.WORD .S.	
8464	024206		MOVE # 0, MPR	;LOAD MAR+HI.
8465		000003	MICPC=MICPC+1	
8466	024206	004000	.WORD .S.	
8467	024210		MOVE BREG, SPAD <1>	;SET THE PARALLEL ADDRESS.
8468		000004	MICPC=MICPC+1	
8469	024210	063221	.WORD .S.	
8470	024212		MOVE BREG, SPAD <2>	;SET THE PARALLEL ADDRESS.
8471		000005	MICPC=MICPC+1	
8472	024212	063222	.WORD .S.	
8473	024214		MOVE SPAD <1>, MEM	;SET THE ADDRESS IN MEMORY.
8474		000006	MICPC=MICPC+1	
8475	024214	062601	.WORD .S.	
8476	024216		MOVE MEM, BREG	;PUT FOUND IN BREG.
8477		000007	MICPC=MICPC+1	
8478	024216	040620	.WORD .S.	
8479	024220		SIFEQ BREG, SPAD <1> 2S	;BRANCH IF GOOD...
8480				
8481				
8482	024220		SUB2C SPAD <1>, BREG, NOP	
8483		000010	MICPC=MICPC+1	
8484	024220	060361	.WORD .S.	
8485	024222		BZ 2S	
8486		000011	MICPC=MICPC+1	
8487	024222	101424	.WORD .S.	
8488	024224		MOVE MEM, OUT1 <CSR5>	;LOAD GOOD DATA...
8489		000012	MICPC=MICPC+1	
8490	024224	041225	.WORD .S.	
8491	024226		MOVE SPAD <1>, BREG	;LOAD BAD DATA...
8492		000013	MICPC=MICPC+1	
8493	024226	060601	.WORD .S.	
8494	024230		MOVE BREG, OUT1 <CSR4>	;
8495		000014	MICPC=MICPC+1	
8496	024230	061224	.WORD .S.	
8497	024232		MOVE # 7, BREG	;SET THE ERROR TYPE...
8498		000015	MICPC=MICPC+1	
8499	024232	000407	.WORD .S.	
8500	024234		MOVE BREG, OUT1 <CSR3>	;
8501		000016	MICPC=MICPC+1	
8502	024234	061223	.WORD .S.	
8503	024236		MOVE SPAD <1>, BREG	;LOAD ADDRESS.
8504		000017	MICPC=MICPC+1	
8505	024236	060601	.WORD .S.	
8506	024240		MOVE BREG, OUT1 <CSR7>	;
8507		000020	MICPC=MICPC+1	
8508	024240	061227	.WORD .S.	
8509	024242		CALL ERROR1	;REPORT ERROR...
8510	024242		MOVE # <MICPC+3>, BREG	
8511		000021	MICPC=MICPC+1	
8512	024242	000423	.WORD .S.	
8513	024244		SBR ERROR1	
8514		000022	MICPC=MICPC+1	
8515	024244	104401	.WORD .S.	

KMC11 MAIN MEMORY TESTS

8516	024246		SBR	IS		;LOOP ON ERROR...
8517		000023	NICPC=NICPC+1			
8518	024246	100406	.WORD	.S.		
8519	024250		25: CALL	SCP11		;SCOPE ON DATA SET?
8520	024250		MOVE	# <NICPC+3>,BREG		
8521		000024	NICPC=NICPC+1			
8522	024250	000426	.WORD	.S.		
8523	024252		SBR	SCP11		
8524		000025	NICPC=NICPC+1			
8525	024252	104430	.WORD	.S.		
8526	024254		SBR	IS		;YES, DO IT...
8527		000026	NICPC=NICPC+1			
8528	024254	100406	.WORD	.S.		
8529	024256		SINC	SPAD <1>,	,MLRLD	;INCREMENT THE ADDRESS.
8530		000027	NICPC=NICPC+1			
8531	024256	073061	.WORD	.S.!.DSP!MLRLD		
8532	024260		SADC	SPAD <2>,	,MPLRD	;INCREMENT PAGE ADDRESS.
8533		000030	NICPC=NICPC+1			
8534	024260	067102	.WORD	.S.!.DSP!MPLRD		
8535	024262		MOVE	# 4,BREG		;CHECK IF DONE?
8536		000031	NICPC=NICPC+1			
8537	024262	000404	.WORD	.S.		
8538	024264		SIFHI	BREG,SPAD <2>	IS	;NO, THEN CONTINUE...
8539						
8540	024264		SUBIC	SPAD <2>,BREG,NOP		
8541		000032	NICPC=NICPC+1			
8542	024264	060342	.WORD	.S.		
8543	024266		BC	648		
8544		000033	NICPC=NICPC+1			
8545	024266	101035	.WORD	.S.		
8546	024270		SBR	IS		
8547		000034	NICPC=NICPC+1			
8548	024270	100406	.WORD	.S.		
8549	024272		648: MOVE	# 0,BREG		;PREPARE TO CHECK IT.
8550	024272		35: NICPC=NICPC+1			
8551		000035	.WORD	.S.		
8552	024272	000400	MOVE	# 0,MLR		;LOAD MAR+LO.
8553	024274		NICPC=NICPC+1			
8554		000036	.WORD	.S.		
8555	024274	010000	MOVE	# 0,MPLR		;LOAD MAR+HI.
8556	024276		NICPC=NICPC+1			
8557		000037	.WORD	.S.		
8558	024276	004000	MOVE	BREG,SPAD <1>		;SET THE PARALLEL ADDRESS.
8559	024300		NICPC=NICPC+1			
8560		000040	.WORD	.S.		
8561	024300	063221	MOVE	BREG,SPAD <2>		;SET THE PARALLEL ADDRESS.
8562	024302		NICPC=NICPC+1			
8563		000041	.WORD	.S.		
8564	024302	063222	45: MOVE	MEM,BREG		;PUT "FOUND" IN BREG.
8565	024304		NICPC=NICPC+1			
8566		000042	.WORD	.S.		
8567	024304	040620	SIFEQ	BREG,SPAD <1>	55	;BRANCH IF GOOD...
8568	024306					
8569						
8570						
8571						

KMC11 MAIN MEMORY TESTS

8572	024306	000043	SUBC SPAD (1),BREG,NOP	
8573		060361	MICPC=MICPC+1	
8574	024306		.WORD .S.	
8575	024310		BZ SS	
8576		000044	MICPC=MICPC+1	
8577	024310	101455	.WORD .S.	
8578	024312		MOVE MEN,OUT1 (CSRS)	;
8579		000045	MICPC=MICPC+1	
8580	024312	041225	.WORD .S.	
8581	024314		MOVE SPAD (1),BREG	;
8582		000046	MICPC=MICPC+1	
8583	024314	060601	.WORD .S.	
8584	024316		MOVE BREG,OUT1 (CSR4)	;GOOD DATA...
8585		000047	MICPC=MICPC+1	
8586	024316	061224	.WORD .S.	
8587	024320		MOVE # 7,BREG	;SET ERROR TYPE.
8588		000050	MICPC=MICPC+1	
8589	024320	000407	.WORD .S.	
8590	024322		MOVE BREG,OUT1 (CSR3)	;
8591		000051	MICPC=MICPC+1	
8592	024322	061223	.WORD .S.	
8593	024324		CALL EROR1	;REPORT ERROR.
8594	024324		MOVE # <MICPC+3>,BREG	
8595		000052	MICPC=MICPC+1	
8596	024324	000454	.WORD .S.	
8597	024326		SBR EROR1	
8598		000053	MICPC=MICPC+1	
8599	024326	104401	.WORD .S.	
8600	024330		SBR 4\$;LOOP ON ERROR.
8601		000054	MICPC=MICPC+1	
8602	024330	100442	.WORD .S.	
8603	024332		CALL SCP11	;SCOPE THE DATA??
8604	024332		MOVE # <MICPC+3>,BREG	
8605		000055	MICPC=MICPC+1	
8606	024332	000457	.WORD .S.	
8607	024334		SBR SCP11	
8608		000056	MICPC=MICPC+1	
8609	024334	104430	.WORD .S.	
8610	024336		SBR 4\$;YES, SCOPE IT.
8611		000057	MICPC=MICPC+1	
8612	024336	100442	.WORD .S.	
8613	024340		SINC SPAD (1), ,MLRLD	;INCREMENT MEMORY ADDRESS.
8614		000060	MICPC=MICPC+1	
8615	024340	073061	.WORD .S.!.DSP!MLRLD	
8616	024342		SADC SPAD (2), ,MPLRLD	;INCREMENT MEMORY ADDRESS.
8617		000061	MICPC=MICPC+1	
8618	024342	067102	.WORD .S.!.DSP!MPLRLD	
8619	024344		MOVE # 4,BREG	;CHECK IF DONE??
8620		000062	MICPC=MICPC+1	
8621	024344	000404	.WORD .S.	
8622	024346		SIFHI BREG,SPAD (2) 4\$;NO, THEN CONTINUE...
8623				
8624				
8625	024346		SUBC SPAD (2),BREG,NOP	
8626		000063	MICPC=MICPC+1	
8627	024346	060342	.WORD .S.	

KMCI1 MAIN MEMORY TESTS

```

8629 024350 000064 BC 658
8630 024350 101066 MICPC=MICPC+1
8631 024352 .WORD .S.
8632 000065 SBR 48
8633 024352 100442 MICPC=MICPC+1
8634 024354 .WORD .S.
8635 024354 658: CALL SCPE ;SCOPE THE TEST...
8636 024354 MOVE # <MICPC+3>,BREG
8637 000066 MICPC=MICPC+1
8638 024354 000470 .WORD .S.
8639 024356 SBR SCPE
8640 000067 MICPC=MICPC+1
8641 024356 104454 .WORD .S.
8642 024360 SBR 68 ;DO THE NEXT ITERATION.
8643 000070 MICPC=MICPC+1
8644 024360 100400 .WORD .S.
8645 024362 SMEM31
8646 024362 SXZ
8647
8648
8649
8650
8651
8652
8653 024362 SXZ ;***** TEST 41 *****
8654 ;* MAR TEST.
8655 ;* PERFORM DUAL ADDRESSING TEST.
8656 ;* USING MAR AUTO-INC FEATURE.
8657
8658
8659 024362 STSTN ;*****
8660 024370 012737 000041 001202 TST41: ; TEST 41
8661 024370 012737 024544 001442 MOV #41,STSTN ; LOAD THE NO. OF THIS TEST
8662 024376 004737 035536 MOV #TST42,NEXT ; POINT TO THE START OF NEXT TEST.
8663 024402 024416 JSR PC,LDRWNT ;RI CONTAINS BASE KMCI1 ADDRESS
8664 024404 104022 MCT41 ;LOAD-VERIFY-WAIT.
8665 024406 012706 001200 ERROR 22 ;TIME OUT ERROR...
8666 024412 000177 155024 MOV #STACK,SP ;RESET STACK...
8667 024416 MCT41: JMP #NEXT ;GO TO NEXT TEST...
8668 024416 15: MOVE # 0,BREG ;START WITH ZERO.
8669 000000 MICPC=MICPC+1
8670 024416 000400 .WORD .S.
8671 024420 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES.
8672 000001 MICPC=MICPC+1
8673 024420 063236 .WORD .S.
8674 024422 MOVE # 0,MLR ;LOAD MAR+LO.
8675 000002 MICPC=MICPC+1
8676 024422 010000 .WORD .S.
8677 024424 MOVE # 0,MPR ;LOAD MAR+HI.
8678 000003 MICPC=MICPC+1
8679 024424 004000 .WORD .S.
8680 024426 MOVE BREG,SPAD <1> ;SET PARALLEL ADDRESS.
8681 000004 MICPC=MICPC+1
8682 024426 063221 .WORD .S.
8683 024430 MOVE BREG,SPAD <2> ;SET PARALLEL ADDRESS.

```

8684		000005		NICPC=NICPC+1	
8685	024430	063222		.WORD .S.	
8686	024432			MOVE # 4, BREG	
8687		000006		NICPC=NICPC+1	
8688	024432	000404		.WORD .S.	
8689	024434		2S:	MOVE SPAD <1>, MEM, MARINC	;LOAD DATA IN MEMORY.
8690		000007		NICPC=NICPC+1	
8691	024434	076601		.WORD .S.	
8692	024436			SINC SPAD <1>	;UPDATE ADDRESS.
8693		000010		NICPC=NICPC+1	
8694	024436	063071		.S. DSP	
8695	024438			SADC SPAD <2>	;UPDATE ADDRESS.
8696	024440			C=NICPC+1	
8697		000011		DSP	
8698	024440	063071		II BREG, SPAD <2> 2S	;BRANCH IF NOT DONE...
8699					
8700					
8701					
8702					
8703					
8704					
8705					
8706					
8707					
8708					
8709					
8710					
8711					
8712					
8713					
8714					
8715					
8716	024452	010000		MOVE # 0, BREG	;START WITH 0 AGAIN.
8717	024454			NICPC=NICPC+1	
8718		000017		.WORD .S.	
8719	024454	004000		MOVE # 0, MLR	;LOAD MAR+LO.
8720				NICPC=NICPC+1	
8721		000020		.WORD .S.	
8722	024456	063221		MOVE BREG, SPAD <1>	;SET PARALLEL ADDRESS.
8723	024460			NICPC=NICPC+1	
8724		000021		.WORD .S.	
8725	024460	063222		MOVE BREG, SPAD <2>	;SET PARALLEL ADDRESS.
8726	024462			NICPC=NICPC+1	
8727		000022		.WORD .S.	
8728	024462	054620	3S:	MOVE MEM, BREG, MARINC	;GET THE CONTENTS OF MEMORY
8729	024464			NICPC=NICPC+1	
8730				.WORD .S.	
8731				SIFE0 BREG, SPAD <1> 4S	;BRANCH IF GOOD.
8732	024464				
8733		000023			
8734	024464	060361		SUBC SPAD <1>, BREG, NOP	
8735	024466			NICPC=NICPC+1	
8736				.WORD .S.	
8737		000024		BZ 4S	
8738	024466	101436		NICPC=NICPC+1	
8739	024470	000025		.WORD .S.	
				MOVE MEM, OUT1 <CSRS>	;GOOD DATA.
				NICPC=NICPC+1	

KMC11 MAIN MEMORY TESTS

8740	024470	041225	.WORD	.S.		
8741	024472		MOVE	SPAD (1),BREG		;BAD DATA.
8742		000026	NICPC=NICPC+1			
8743	024472	060601	.WORD	.S.		
8744	024474		MOVE	BREG OUT1 (CSR4)		;
8745		000027	NICPC=NICPC+1			
8746	024474	061224	.WORD	.S.		
8747	024476		MOVE	B 10,BREG		;SET THE ERROR TYPE.
8748		000030	NICPC=NICPC+1			
8749	024476	000410	.WORD	.S.		
8750	024500		MOVE	BREG OUT1 (CSR3)		;
8751		000031	NICPC=NICPC+1			
8752	024500	061223	.WORD	.S.		
8753	024502		CALL	ENOR		;REPORT ERROR.
8754	024502		MOVE	B (NICPC+3),BREG		
8755		000032	NICPC=NICPC+1			
8756	024502	000434	.WORD	.S.		
8757	024504		SEN	ENOR		
8758		000033	NICPC=NICPC+1			
8759	024504	104400	.WORD	.S.		
8760	024506		MOVE	SPAD (1),MLR		;LOOP ON ERROR.
8761		000034	NICPC=NICPC+1			
8762	024506	070201	.WORD	.S.		
8763	024510		SEN	35		;LOOP ON ERROR.
8764		000035	NICPC=NICPC+1			
8765	024510	100422	.WORD	.S.		
8766	024512		CALL	SCP1		;SCOPE THE DATA...
8767	024512		MOVE	B (NICPC+3),BREG		
8768		000036	NICPC=NICPC+1			
8769	024512	000440	.WORD	.S.		
8770	024514		SEN	SCP1		
8771		000037	NICPC=NICPC+1			
8772	024514	104427	.WORD	.S.		
8773	024516		MOVE	SPAD (1),MLR		;RESTORE POINTER...
8774		000040	NICPC=NICPC+1			
8775	024516	070201	.WORD	.S.		
8776	024520		SEN	35		;LOOP ON DATA...
8777		000041	NICPC=NICPC+1			
8778	024520	100422	.WORD	.S.		
8779	024522		STNC	SPAD (1)		;UPDATE ADDRESS.
8780		000042	NICPC=NICPC+1			
8781	024522	063061	.WORD	.S.!.DSP		
8782	024524		STNC	SPAD (2)		;UPDATE ADDRESS.
8783		000043	NICPC=NICPC+1			
8784	024524	063102	.WORD	.S.!.DSP		
8785	024526		MOVE	B 4,BREG		;CHECK IF DONE.
8786		000044	NICPC=NICPC+1			
8787	024526	000404	.WORD	.S.		
8788	024530		SIFHI	BREG,SPAD (2) 35		;IF NOT THEN CONTINUE...
8789						
8790						
8791	024530		SUBIC	SPAD (2),BREG,NOP		
8792		000045	NICPC=NICPC+1			
8793	024530	060342	.WORD	.S.		
8794	024532		BC	655		
8795		000046	NICPC=NICPC+1			

KMC11 MAIN MEMORY TESTS

8796	024532	101050		
8797	024534			
8798		000047		
8799	024534	100422		
8800	024536			
8801	024536			
8802	024536			
8803		000050		
8804	024536	000452		
8805	024540			
8806		000051		
8807	024540	104454		
8808	024542			
8809		000052		
8810	024542	100400		
8811	024544			
8812	024544			
8813				
8814				
8815				
8816				
8817				
8818				
8819	024544			
8820				
8821	024544			
8822				
8823				
8824	024544	012737	000042	001202
8825	024552	012737	024700	001442
8826				
8827	024560	004737	035536	
8828	024564	024600		
8829	024566	104022		
8830	024570	012706	001200	
8831	024574	000177	154642	
8832				
8833	024600			
8834	024600			
8835		000000		
8836	024600	000400		
8837	024602			
8838		000001		
8839	024602	063236		
8840	024604			
8841		000002		
8842	024604	010000		
8843	024606			
8844		000003		
8845	024606	004000		
8846	024610			
8847		000004		
8848	024610	063220		
8849	024612			
8850		000005		
8851	024612	063221		

```

WORD .S.
SER .S.
MICPC=MICPC+1
.WORD .S.
65S:
CALL SCPE ;SCOPE THE TEST.
MOVE # (MICPC+3), BREG
MICPC=MICPC+1
.WORD .S.
SER SCPE
MICPC=MICPC+1
.WORD .S.
SER IS ;DO THE NEXT ITERATION...
MICPC=MICPC+1
.WORD .S.
SALUTO
SXZ

;***** TEST 42 *****
;#ALU C BIT TEST
;#TEST THAT ADD OF 377 AND 1 WILL SET TAG CBIT.
;#THEN CHECK IF C BIT CLEARS
SXZ

;*****
STSTN
; TEST 42
TST42: MOV #42, STSTN ; LOAD THE NO. OF THIS TEST
MOV #ST43, NEXT ; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
;LOAD-VERIFY-WAIT.
JSR PC, LDVWNT
MCT42 ERROR 22 ;TIME OUT ERROR...
MOV #STACK, SP ;RESET STACK...
JMP @NEXT ;GO TO NEXT TEST...

MCT42: IS: MOVE # 0, BREG ;SET TO CLEAR REQUIRED LOCATIONS.
MICPC=MICPC+1
.WORD .S.
MOVE BREG, SPAD <16> ;FOR RETURN ADDRESS PURPOSES.
MICPC=MICPC+1
.WORD .S.
MOVE # 0, MLR ;
MICPC=MICPC+1
.WORD .S.
MOVE # 0, MPR ;
MICPC=MICPC+1
.WORD .S.
MOVE BREG, SPAD <0> ;CLEAR SPAD 0.
MICPC=MICPC+1
.WORD .S.
MOVE BREG, SPAD <1> ;CLEAR SPAD 1.
MICPC=MICPC+1
.WORD .S.

```

KMC11 ALU TESTS

8852	024614		MOVE BREG SPAD (2)	;CLEAR SPAD 2.
8853		000006	MICPC=MICPC+1	
8854	024614	063222	.WORD .S.	
8855	024616		MOVE # 377, MEM	;
8856		000007	MICPC=MICPC+1	
8857	024616	002777	.WORD .S.	
8858	024620		MOVE MEM, SPAD (1)	;LOAD 377 IN SPAD 1.
8859		000010	MICPC=MICPC+1	
8860	024620	043221	.WORD .S.	
8861	024622		SINC SPAD (1)	;ADD 1 TO IT.
8862		000011	MICPC=MICPC+1	
8863	024622	063061	.WORD .S.!.DSP	
8864	024624		SADC SPAD (2)	;GET THE CARRY BIT..
8865		000012	MICPC=MICPC+1	
8866	024624	063102	.WORD .S.!.DSP	
8867	024626		MOVE SPAD (2), BREG	;IS CARRY SET??
8868		000013	MICPC=MICPC+1	
8869	024626	060602	.WORD .S.	
8870	024630		BBO # 25	;YES SCOPE THE TEST DATA.
8871		000014	MICPC=MICPC+1	
8872	024630	102022	.WORD .S.	
8873	024632		65: MOVE # 21 MEM	;GOOD DATA.
8874	024632		MICPC=MICPC+1	;TYPE OF ERROR.
8875		000015	.WORD .S.	
8876	024632	002421	MOVE MEM, OUT1 (CSR3)	;
8877	024634		MICPC=MICPC+1	
8878		000016	.WORD .S.	
8879	024634	041223	CALL EROR1	;REPORT ERROR...
8880	024636		MOVE # (MICPC+3), BREG	
8881	024636		MICPC=MICPC+1	
8882		000017	.WORD .S.	
8883	024636	000421	SBR EROR1	
8884	024640		MICPC=MICPC+1	
8885		000020	.WORD .S.	
8886	024640	104401	SBR IS	;LOOP ON ERROR...
8887	024642		MICPC=MICPC+1	
8888		000021	.WORD .S.	
8889	024642	100400	25: CALL SCP11	;SCOPE THE DATA??
8890	024644		MOVE # (MICPC+3), BREG	
8891	024644		MICPC=MICPC+1	
8892		000022	.WORD .S.	
8893	024644	000424	SBR SCP11	
8894	024646		MICPC=MICPC+1	
8895		000023	.WORD .S.	
8896	024646	104430	SBR IS	;YES, DO IT...
8897	024650		MICPC=MICPC+1	
8898	024650	100400	.WORD .S.	
8899	024652		35: SDEC SPAD (2)	;RESET THE SPAD (2).
8900		000025	MICPC=MICPC+1	
8901	024652	063162	.WORD .S.!.DSP	
8902	024654		MOVE # 377, MEM	;LOAD 377 IN MEMORY.
8903		000026	MICPC=MICPC+1	
8904	024654	002777	.WORD .S.	
8905	024656		MOVE MEM, SPAD (1)	;LOAD IN SPAD. (1).
8906		000027	MICPC=MICPC+1	

KMC11 ALU TESTS

8908	024656	043221				WORD	S		
8909	024660					SINC	SPAD <1>		;ADD 1 TO IT.
8910		000030				MICPC=MICPC+1			
8911	024660	063061			.WORD	.S...DSP			
8912	024662					SDEC	SPAD <1>		;SUBTRACT 1 OUT OF IT.
8913		000031				MICPC=MICPC+1			
8914	024662	063161			.WORD	.S...DSP			
8915	024664					SADC	SPAD <0>		;GET THE CARRY BIT.
8916		000032				MICPC=MICPC+1			
8917	024664	063100			.WORD	.S...DSP			
8918	024666				45:	MOVE	SPAD <0>,BREG		;WAS IT RESET?
8919		000033				MICPC=MICPC+1			
8920	024666	060600				WORD	S		
8921	024670					BBO	6S		;NO, REPORT ERROR.
8922		000034				MICPC=MICPC+1			
8923	024670	102015			.WORD	S			
8924	024672					CALL	SCPE		;SCOPE THE TEST.
8925	024672					MOVE	# <MICPC+3>,BREG		
8926		000035				MICPC=MICPC+1			
8927	024672	000437			.WORD	S			
8928	024674					SBR	SCPE		
8929		000036				MICPC=MICPC+1			
8930	024674	104454			.WORD	S			
8931	024676					SBR	IS		;DO THE NEXT ITERATION.
8932		000037				MICPC=MICPC+1			
8933	024676	100400			.WORD	S			
8934	024700				SALUT1	0,<SEL B & SEL A>,MOVE,0,0,-1,-1,0,0,-1,-1,125,125,252,252,125,125,252,252,,0			
8935	024700				SXZ				
8936									
8937									
8938									
8939									
8940									
8941									
8942									
8943									
8944									
8945									
8946									
8947									
8948									
8949									
8950									
8951									
8952									
8953									
8954									
8955									
8956									
8957									
8958									
8959									
8960									
8961									
8962									
8963									
8964									
8965									
8966									
8967									
8968									
8969									
8970									
8971									
8972									
8973									
8974									
8975									
8976									
8977									
8978									
8979									
8980									
8981									
8982									
8983									
8984									
8985									
8986									
8987									
8988									
8989									
8990									
8991									
8992									
8993									
8994									
8995									
8996									
8997									
8998									
8999									
9000									

```

***** TEST 43 *****
#ALU TEST
#TEST OF ALU FUNCTION SEL B & SEL A WITH C BIT CLEARED.
#TEST OF ALU FUNCTION SEL B & SEL A WITH C BIT SET.
#ALU FUNCTION (B)
#LOAD MAIN MEMORY 16 WORDS OF DATA.
#PERFORM THE FUNCTION, VERIFY THE RESULTS..

```

TEST 43

```

TST43: MOV #43,STSTN ; LOAD THE NO. OF THIS TEST
MOV #TST44,NEXT ; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
;LOAD-VERIFY-WAIT.
MCT43: JSR PC,LVWRT
ERROR 22 ;TIME OUT ERROR...
MOV #STACK,SP ;RESET STACK.
JMP @NEXT ;GO TO NEXT TEST...
15: MOVE #0,MLR ;SET MAR+LO.
MICPC=MICPC+1
WORD S
MOVE #0,MPR ;SET MAR+HI.

```


8964		000001	MICPC=MICPC+1	
8965	024736	004000	.WORD .S.	
8966	024740		MOVE # 0 BREG	
8967		000002	MICPC=MICPC+1	
8968	024740	000400	.WORD .S.	
8969	024742		MOVE BREG SPAD <16>	;FOR RETURN ADDRESS...
8970		000003	MICPC=MICPC+1	
8971	024742	063236	.WORD .S.	
8972	024744		MOVE BREG SPAD <0>	;
8973		000004	MICPC=MICPC+1	
8974	024744	063220	.WORD .S.	
8975	024746		MOVE BREG SPAD <1>	;
8976		000005	MICPC=MICPC+1	
8977	024746	063221	.WORD .S.	
8978	024750		MOVE BREG SPAD <2>	;
8979		000006	MICPC=MICPC+1	
8980	024750	063222	.WORD .S.	
8981	024752		SDEC SPAD <2>	;
8982		000007	MICPC=MICPC+1	
6783	024752	063162	.WORD .S.!.DSP	
8984	024754		MOVE BREG SPAD <4>	;
8985		000010	MICPC=MICPC+1	
8986	024754	063224	.WORD .S.	
8987	024756		MOVE # 0 MEM MARINC	;LOAD THE DATA IN MEMORY.
8988		000011	MICPC=MICPC+1	
8989	024756	016400	.WORD .S.	
8990	024760		MOVE # 0 MEM MARINC	;LOAD THE DATA IN MEMORY
8991		000012	MICPC=MICPC+1	
8992	024760	016400	.WORD .S.	
8993	024762		MOVE # 0 MEM MARINC	
8994		000013	MICPC=MICPC+1	
8995	024762	016400	.WORD .S.	
8996	024764		MOVE # 0 MEM MARINC	;RESULT WITH C BIT SET.
8997		000014	MICPC=MICPC+1	
8998	024764	016400	.WORD .S.	
8999	024766		MOVE # -1 MEM MARINC	;LOAD THE DATA IN MEMORY.
9000		000015	MICPC=MICPC+1	
9001	024766	016777	.WORD .S.	
9002	024770		MOVE # 0 MEM MARINC	;LOAD THE DATA IN MEMORY.
9003		000016	MICPC=MICPC+1	
9004	024770	016400	.WORD .S.	
9005	024772		MOVE # -1 MEM MARINC	
9006		000017	MICPC=MICPC+1	
9007	024772	016777	.WORD .S.	
9008	024774		MOVE # -1 MEM MARINC	;RESULT WITH C BIT SET.
9009		000020	MICPC=MICPC+1	
9010	024774	016777	.WORD .S.	
9011	024776		MOVE # 0 MEM MARINC	;LOAD THE DATA IN MEMORY.
9012		000021	MICPC=MICPC+1	
9013	024776	016400	.WORD .S.	
9014	025000		MOVE # -1 MEM MARINC	;LOAD THE DATA IN MEMORY.
9015		000022	MICPC=MICPC+1	
9016	025000	016777	.WORD .S.	
9017	025002		MOVE # 0 MEM MARINC	
9018		000023	MICPC=MICPC+1	
9019	025002	016400	.WORD .S.	

9020	025004	000024	MOVE # 0, MEM MARINC ;RESULT WITH C BIT SET.
9021		016400	NICPC=NICPC+1
9022	025004	016400	.WORD .S.
9023	025006		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY
9024		000025	NICPC=NICPC+1
9025	025006	016777	.WORD .S.
9026	025010		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.
9027		000026	NICPC=NICPC+1
9028	025010	016777	.WORD .S.
9029	025012		MOVE # -1, MEM MARINC
9030		000027	NICPC=NICPC+1
9031	025012	016777	.WORD .S.
9032	025014		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.
9033		000030	NICPC=NICPC+1
9034	025014	016777	.WORD .S.
9035	025016		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
9036		000031	NICPC=NICPC+1
9037	025016	016525	.WORD .S.
9038	025020		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
9039		000032	NICPC=NICPC+1
9040	025020	016525	.WORD .S.
9041	025022		MOVE # 125, MEM MARINC
9042		000033	NICPC=NICPC+1
9043	025022	016525	.WORD .S.
9044	025024		MOVE # 125, MEM MARINC ;RESULT WITH C BIT SET.
9045		000034	NICPC=NICPC+1
9046	025024	016525	.WORD .S.
9047	025026		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
9048		000035	NICPC=NICPC+1
9049	025026	016652	.WORD .S.
9050	025030		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
9051		000036	NICPC=NICPC+1
9052	025030	016525	.WORD .S.
9053	025032		MOVE # 252, MEM MARINC
9054		000037	NICPC=NICPC+1
9055	025032	016652	.WORD .S.
9056	025034		MOVE # 252, MEM MARINC ;RESULT WITH C BIT SET.
9057		000040	NICPC=NICPC+1
9058	025034	016652	.WORD .S.
9059	025036		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
9060		000041	NICPC=NICPC+1
9061	025036	016525	.WORD .S.
9062	025040		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY
9063		000042	NICPC=NICPC+1
9064	025040	016652	.WORD .S.
9065	025042		MOVE # 125, MEM MARINC
9066		000043	NICPC=NICPC+1
9067	025042	016525	.WORD .S.
9068	025044		MOVE # 125, MEM MARINC ;RESULT WITH C BIT SET.
9069		000044	NICPC=NICPC+1
9070	025044	016525	.WORD .S.
9071	025046		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
9072		000045	NICPC=NICPC+1
9073	025046	016652	.WORD .S.
9074	025050		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
9075		000046	NICPC=NICPC+1

9076	025050	016652	.WORD .S.	
9077	025052		MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY.
9078		000047	MICPC=MICPC+1	
9079	025052	016652	.WORD .S.	
9080	025054		MOVE # 252, MEM MARINC	;RESULT WITH C BIT SET.
9081		000050	MICPC=MICPC+1	
9082	025054	016652	.WORD .S.	
9083	025056		MOVE # 7, SPAD <7>	;SET THE COUNT.
9084		000051	MICPC=MICPC+1	
9085	025056	003007	.WORD .S.	
9086				
9087	025060		MOVE # 0, MLR	;MAR+0.
9088		000052	MICPC=MICPC+1	
9089	025060	010000	.WORD .S.	
9090				
9091	025062		25: MOVE # 0, BREG	;
9092		000053	MICPC=MICPC+1	
9093	025062	000400	.WORD .S.	
9094	025064		SADD SPAD <1>, BREG	;CLEAR C BIT.
9095		000054	MICPC=MICPC+1	
9096	025064	060401	.WORD .S. DD	
9097	025066		MOVE MEM, SPAD <0> MARINC	;GET THE FIRST OPERAND.
9098		000055	MICPC=MICPC+1	
9099	025066	057220	.WORD .S.	
9100	025070		MOVE SPAD <0>, BR.SP, MARINC	;
9101		000056	MICPC=MICPC+1	
9102	025070	077600	.WORD .S.	
9103	025072		SIFEQ MEM, SPAD <0> 3S	;BRANCH IF GOOD.
9104				
9105				
9106	025072		SUBPC SPAD <0>, MEM, NOP	
9107		000057	MICPC=MICPC+1	
9108	025072	040360	.WORD .S.	
9109	025074		BZ 3S	
9110		000060	MICPC=MICPC+1	
9111	025074	101473	.WORD .S.	
9112	025076		MOVE MEM, OUT1 <4>	;GOOD DATA
9113		000061	MICPC=MICPC+1	
9114	025076	041224	.WORD .S.	
9115	025100		MOVE BREG, OUT1 <5>	;BAD DATA.
9116		000062	MICPC=MICPC+1	
9117	025100	061225	.WORD .S.	
9118	025102		MOVE # 15, BREG	;SET TYPE OF ERROR.
9119		000063	MICPC=MICPC+1	
9120	025102	000415	.WORD .S.	
9121	025104		MOVE BREG, OUT1 <3>	;SET TYPE OF ERROR.
9122		000064	MICPC=MICPC+1	
9123	025104	061223	.WORD .S.	
9124	025106		MOVE # 11, BREG	;LOAD FUNCTION CODE...
9125		000065	MICPC=MICPC+1	
9126	025106	000411	.WORD .S.	
9127	025110		MOVE BREG, OUT1 <CSR7>	;LOAD IT...
9128		000066	MICPC=MICPC+1	
9129	025110	061227	.WORD .S.	
9130	025112		CALL EROR	;ALU SEL B & SEL A ERROR...
9131	025112		MOVE # <MICPC+3>, BREG	

9132		000067	MICPC=MICPC+1	
9133	025112	000471	.WORD .S.	
9134	025114		SBR EROR	
9135		000070	MICPC=MICPC+1	
9136	025114	104400	.WORD .S.	
9137	025116		MOVE SPAD <4>,MLR	;RESET DATA POINTER...
9138		000071	MICPC=MICPC+1	
9139	025116	070204	.WORD .S.	
9140	025120		SBR 25	;LOOP ON ERROR...
9141		000072	MICPC=MICPC+1	
9142	025120	100453	.WORD .S.	
9143	025122		CALL SCP1	
9144	025122		MOVE # <MICPC+3>,BREG	
9145		000073	MICPC=MICPC+1	
9146	025122	000475	.WORD .S.	
9147	025124		SBR SCP1	
9148		000074	MICPC=MICPC+1	
9149	025124	104427	.WORD .S.	
9150	025126		MOVE SPAD <4>,MLR	;
9151		000075	MICPC=MICPC+1	
9152	025126	070204	.WORD .S.	
9153	025130		SBR 25	;SCOPE THE DATA....
9154		000076	MICPC=MICPC+1	
9155	025130	100453	.WORD .S.	
9156	025132		MOVE SPAD <4>,MLR	;RESET DATA POINTER...
9157		000077	MICPC=MICPC+1	
9158	025132	070204	.WORD .S.	
9159	025134		MOVE MEM SPAD <0>,MARINC	;GET FIRST OPRAND...
9160		000100	MICPC=MICPC+1	
9161	025134	057220	.WORD .S.	
9162	025136		MOVE # 377,BREG	;
9163		000101	MICPC=MICPC+1	
9164	025136	000777	.WORD .S.	
9165	025140		SADD SPAD <2>,BREG	;SET C BIT...
9166		000102	MICPC=MICPC+1	
9167	025140	060402	.WORD .S.!.DO	
9168	025142		MOVE SPAD <0>,BR.SP,MARINC	;SP 0 & BR = SEL B & SEL A
9169		000103	MICPC=MICPC+1	
9170	025142	077600	.WORD .S.	
9171	025144		MOVE MEM SPAD <3>,MARINC	;DUMMY INSTR, TO MARINC.
9172		000104	MICPC=MICPC+1	
9173	025144	057223	.WORD .S.	
9174	025146		SIFEQ MEM,SPAD <0> 95	;BR IF GOOD...
9175				
9176				
9177	025146		SUB2C SPAD <0>,MEM,NOP	
9178		000105	MICPC=MICPC+1	
9179	025146	040360	.WORD .S.	
9180	025150		BZ 95	
9181		000106	MICPC=MICPC+1	
9182	025150	101521	.WORD .S.	
9183	025152		MOVE MEM,OUT1 <CSR4>	;GOOD DATA.
9184		000107	MICPC=MICPC+1	
9185	025152	041224	.WORD .S.	
9186	025154		MOVE BREG,OUT1 <CSR5>	;BAD DATA.
9187		000110	MICPC=MICPC+1	

9188	025154	061225	.WORD .S.	
9189	025156		MOVE # 23,BREG	;ERROR TYPE...
9190		000111	MICPC=MICPC+1	
9191	025156	000423	.WORD .S.	
9192	025160		MOVE BREG,OUT1 <CSR3>	;
9193		000112	MICPC=MICPC+1	
9194	025160	061223	.WORD .S.	
9195	025162		MOVE # 11,BREG	;ALU FUNCTION CODE.
9196		000113	MICPC=MICPC+1	
9197	025162	000411	.WORD .S.	
9198	025164		MOVE BREG,OUT1 <CSR7>	;LOAD IT...
9199		000114	MICPC=MICPC+1	
9200	025164	061227	.WORD .S.	
9201	025166		CALL EROR ;REPORT ERROR...	
9202	025166		MOVE # <MICPC+3>,BREG	
9203		000115	MICPC=MICPC+1	
9204	025166	000517	.WORD .S.	
9205	025170		SBR EROR	
9206		000116	MICPC=MICPC+1	
9207	025170	104400	.WORD .S.	
9208	025172		MOVE SPAD <4>,MLR	;RESTORE DATA POINTER.
9209		000117	MICPC=MICPC+1	
9210	025172	070204	.WORD .S.	
9211	025174		SBR 6\$;LOOP ON ERROR...
9212		000120	MICPC=MICPC+1	
9213	025174	100477	.WORD .S.	
9214	025176		CALL SCP1 ;SCOPE THE ERROR...	
9215	025176		MOVE # <MICPC+3>,BREG	
9216		000121	MICPC=MICPC+1	
9217	025176	000523	.WORD .S.	
9218	025200		SBR SCP1	
9219		000122	MICPC=MICPC+1	
9220	025200	104427	.WORD .S.	
9221	025202		MOVE SPAD <4>,MLR	;RESTORE DATA POINTER...
9222		000123	MICPC=MICPC+1	
9223	025202	070204	.WORD .S.	
9224	025204		SBR 6\$;SCOPE THE DATA...
9225		000124	MICPC=MICPC+1	
9226	025204	100477	.WORD .S.	
9227	025206		MOVE # 4,BREG	;UPDATE BACKGROUND POINTER.
9228		000125	MICPC=MICPC+1	
9229	025206	000404	.WORD .S.	
9230	025210		SADD BREG,SPAD <4>,MARINC	;ALSO DATA POINTER.
9231	025210	077004	.WORD .S. !MARINC!.DO	
9232	025212		SDEC SPAD <7>	;IS IT DONE??
9233		000126	MICPC=MICPC+1	
9234	025212	063167	.WORD .S.!.DSP	
9235	025214		BZ 4\$;YES, SCOPE THE TEST.
9236		000127	MICPC=MICPC+1	
9237	025214	101532	.WORD .S.	
9238	025216		SBR 2\$;DO, THE NEXT.
9239		000130	MICPC=MICPC+1	
9240	025216	100453	.WORD .S.	
9241	025220		CALL SCPE ;SCOPE THE TEST...	
9242	025220		MOVE # <MICPC+3>,BREG	
9243		000131	MICPC=MICPC+1	

025220
025222
025222
025224
025224
025226
025226
025226
025226
025226
025234
025242
025246
025250
025254
025256
025262
025264
025266
025270
025270
025272
025274
025274
025276
025276
025300

000533
000132
104454
000133
100400

012737 000044 001202
012737 025554 001442

004737 035536
025262
104022
012706 001200
000177 154160

000000
010000

000001
004000

000002
000400

000003
063236

000004
063220

000005
063221

000006
063222
000007

```
WORD .S.  
SER .S.  
NICPC=NICPC+1  
WORD .S.  
SER .S.  
NICPC=NICPC+1  
WORD .S.  
SALUT1 0,<A OR NOTB>,LORN,-1,-1,-1,-1,0,0,-1,-1,-1,-1,252,252,125,125,-1,-1,<A OR NOTB>  
SXZ
```

;DO THE NEXT ITERATION...

```
***** TEST 44 *****  
*ALU TEST  
*TEST OF ALU FUNCTION A OR NOTB WITH C BIT CLEARED.  
*TEST OF ALU FUNCTION A OR NOTB WITH C BIT SET.  
*ALU FUNCTION (A OR NOTB)  
*LOAD MAIN MEMORY 16 WORDS OF DATA.  
*PERFORM THE FUNCTION, VERIFY THE RESULTS..
```

SXZ

STSTN

; TEST 44

TST44:

```
MOV #44,STSTN  
MOV #TST45,NEXT
```

; LOAD THE NO. OF THIS TEST
; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
;LOAD-VERIFY-WAIT.

JSR

```
PC,LDRMT
```

MCT44

```
ERROR 22
```

;TIME OUT ERROR...

MOV

```
#STACK,SP
```

;RESET STACK...

JMP

```
@NEXT
```

;GO TO NEXT TEST...

MCT44:
IS:

```
MOVE #0,MLR
```

;SET MAR+LO.

```
NICPC=NICPC+1
```

```
.WORD .S.
```

```
MOVE #0,MPR
```

;SET MAR+HI.

```
NICPC=NICPC+1
```

```
.WORD .S.
```

```
MOVE #0,BREG
```

```
NICPC=NICPC+1
```

```
.WORD .S.
```

```
MOVE BREG,SPAD <16>
```

;FOR RETURN ADDRESS...

```
NICPC=NICPC+1
```

```
.WORD .S.
```

```
MOVE BREG,SPAD <0>
```

```
NICPC=NICPC+1
```

```
.WORD .S.
```

```
MOVE BREG,SPAD <1>
```

```
NICPC=NICPC+1
```

```
.WORD .S.
```

```
MOVE BREG,SPAD <2>
```

```
NICPC=NICPC+1
```

```
.WORD .S.
```

```
SDEC SPAD <2>
```

```
NICPC=NICPC+1
```

9300	025300	063162	.WORD	.S.	DSP		
9301	025302		MOVE	# 0	BREG, SPAD <4>		;
9302		000010	MICPC=	MICPC+1			
9303	025302	063224	.WORD	.S.			
9304	025304		MOVE	# 0	MEM MARINC		;LOAD THE DATA IN MEMORY.
9305		000011	MICPC=	MICPC+1			
9306	025304	016400	.WORD	.S.			
9307	025306		MOVE	# 0	MEM MARINC		;LOAD THE DATA IN MEMORY
9308		000012	MICPC=	MICPC+1			
9309	025306	016400	.WORD	.S.			
9310	025310		MOVE	# -1	MEM MARINC		
9311		000013	MICPC=	MICPC+1			
9312	025310	016777	.WORD	.S.			
9313	025312		MOVE	# -1	MEM MARINC		;RESULT WITH C BIT SET.
9314		000014	MICPC=	MICPC+1			
9315	025312	016777	.WORD	.S.			
9316	025314		MOVE	# -1	MEM MARINC		;LOAD THE DATA IN MEMORY.
9317		000015	MICPC=	MICPC+1			
9318	025314	016777	.WORD	.S.			
9319	025316		MOVE	# 0	MEM MARINC		;LOAD THE DATA IN MEMORY.
9320		000016	MICPC=	MICPC+1			
9321	025316	016400	.WORD	.S.			
9322	025320		MOVE	# -1	MEM MARINC		
9323		000017	MICPC=	MICPC+1			
9324	025320	016777	.WORD	.S.			
9325	025322		MOVE	# -1	MEM MARINC		;RESULT WITH C BIT SET.
9326		000020	MICPC=	MICPC+1			
9327	025322	016777	.WORD	.S.			
9328	025324		MOVE	# 0	MEM MARINC		;LOAD THE DATA IN MEMORY.
9329		000021	MICPC=	MICPC+1			
9330	025324	016400	.WORD	.S.			
9331	025326		MOVE	# -1	MEM MARINC		;LOAD THE DATA IN MEMORY.
9332		000022	MICPC=	MICPC+1			
9333	025326	016777	.WORD	.S.			
9334	025330		MOVE	# 0	MEM MARINC		
9335		000023	MICPC=	MICPC+1			
9336	025330	016400	.WORD	.S.			
9337	025332		MOVE	# 0	MEM MARINC		;RESULT WITH C BIT SET.
9338		000024	MICPC=	MICPC+1			
9339	025332	016400	.WORD	.S.			
9340	025334		MOVE	# -1	MEM MARINC		;LOAD THE DATA IN MEMORY
9341		000025	MICPC=	MICPC+1			
9342	025334	016777	.WORD	.S.			
9343	025336		MOVE	# -1	MEM MARINC		;LOAD THE DATA IN MEMORY.
9344		000026	MICPC=	MICPC+1			
9345	025336	016777	.WORD	.S.			
9346	025340		MOVE	# -1	MEM MARINC		
9347		000027	MICPC=	MICPC+1			
9348	025340	016777	.WORD	.S.			
9349	025342		MOVE	# -1	MEM MARINC		;RESULT WITH C BIT SET.
9350		000030	MICPC=	MICPC+1			
9351	025342	016777	.WORD	.S.			
9352	025344		MOVE	# 125	MEM MARINC		;LOAD THE DATA IN MEMORY.
9353		000031	MICPC=	MICPC+1			
9354	025344	016525	.WORD	.S.			
9355	025346		MOVE	# 125	MEM MARINC		;LOAD THE DATA IN MEMORY.

9356		000032	NICPC=NICPC+1	
9357	025376	016525	.WORD .S.	
9358	025360		MOVE # -1, MEM MARINC	
9359		000033	NICPC=NICPC+1	
9360	025360	016777	.WORD .S.	
9361	025362		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.	
9362		000034	NICPC=NICPC+1	
9363	025362	016777	.WORD .S.	
9364	025364		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
9365	025366	000035	NICPC=NICPC+1	
9366		016652	.WORD .S.	
9367		000036	MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
9368	025366	016525	.WORD .S.	
9369	025360		MOVE # 252, MEM MARINC	
9370		000037	NICPC=NICPC+1	
9371	025360	016652	.WORD .S.	
9372	025362		MOVE # 252, MEM MARINC ;RESULT WITH C BIT SET.	
9373		000040	NICPC=NICPC+1	
9374	025362	016652	.WORD .S.	
9375	025364		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
9376		000041	NICPC=NICPC+1	
9377	025364	016525	.WORD .S.	
9378	025366		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY	
9379		000042	NICPC=NICPC+1	
9380	025366	016652	.WORD .S.	
9381	025370		MOVE # 125, MEM MARINC	
9382		000043	NICPC=NICPC+1	
9383	025370	016525	.WORD .S.	
9384	025372		MOVE # 125, MEM MARINC ;RESULT WITH C BIT SET.	
9385		000044	NICPC=NICPC+1	
9386	025372	016525	.WORD .S.	
9387	025374		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
9388		000045	NICPC=NICPC+1	
9389	025374	016652	.WORD .S.	
9390	025376		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
9391		000046	NICPC=NICPC+1	
9392	025376	016652	.WORD .S.	
9393	025400		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.	
9394		000047	NICPC=NICPC+1	
9395	025400	016777	.WORD .S.	
9396	025402		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.	
9397		000050	NICPC=NICPC+1	
9398	025402	016777	.WORD .S.	
9399	025404		MOVE # 7, SPAD (<7>) ;SET THE COUNT.	
9400		000051	NICPC=NICPC+1	
9401	025404	003007	.WORD .S.	
9402				
9403				
9404	025406		MOVE # 0, MLR ;MAR+0.	
9405		000052	NICPC=NICPC+1	
9406	025406	010000	.WORD .S.	
9407				
9408	025410			
9409		000053	25: MOVE # 0, BREG ;	
9410	025410	000400	NICPC=NICPC+1	
9411	025412		.WORD .S.	
			SADD SPAD (<1>, BREG ;CLEAR C BIT.	


```

9412 000054      NICPC=NICPC+1
9413 025412 060401      .WORD      .S.!.DD
9414 025414      MOVE      MEM,SPAD <0> MARINC      ;GET THE FIRST OPERAND.
9415 000055      NICPC=NICPC+1
9416 025414 057220      .WORD      .S.
9417 025416      LOW      MEM,SPAD <0>,BR.SP,MARINC      ;SPAD <0>:=SFUNC.
9418 000056      NICPC=NICPC+1
9419 025416 057640      .WORD      .S.
9420 025420      $IFEQ      MEM,SPAD <0> 3S ;BRANCH IF GOOD.

025420      SUBRC      SPAD <0>,MEM,NOP
000057      NICPC=NICPC+1
025420 040260      .WORD      .S.-
025422      BZ
000060      NICPC=NICPC+1
025422 101473      .WORD      .S.
025424      MOVE      MEM,OUT1 <4>      ;GOOD DATA
000061      NICPC=NICPC+1
025424 041224      .WORD      .S.
025426      MOVE      BREG,OUT1 <5>      ;BAD DATA.
000062      NICPC=NICPC+1
025426 061225      .WORD      .S.
025428      MOVE      #15,BREG      ;SET TYPE OF ERROR.
000063      NICPC=NICPC+1
025428 000416      .WORD      .S.
025430      MOVE      BREG,OUT1 <3>      ;SET TYPE OF ERROR.
000064      NICPC=NICPC+1
025430 051224      .WORD      .S.
025432      MOVE      #12,BREG      ;LOAD FUNCTION CODE...
000065      NICPC=NICPC+1
025432 000412      .WORD      .S.
025434      MOVE      BREG,OUT1 <CSR7>      ;LOAD IT...
000066      NICPC=NICPC+1
025434 051227      .WORD      .S.
025436      CALL      ENDR      ;ALU A OR NOTB ERROR...
000067      NICPC=NICPC+1
025436 000471      .WORD      .S.
025438      MOVE      # <NICPC+3>,BREG
000070      NICPC=NICPC+1
025438 104400      .WORD      .S.
025440      MOVE      SPAD <4>,MLR      ;RESET DATA POINTER...
000071      NICPC=NICPC+1
025440 070804      .WORD      .S.
025442      SR      #25      ;LOOP ON ERROR...
000072      NICPC=NICPC+1
025442 100463      .WORD      .S.
3S: 025444      CALL      SCP1
000073      NICPC=NICPC+1
025444 000475      .WORD      .S.
025446      MOVE      # <NICPC+3>,BREG
000074      NICPC=NICPC+1
025446 104427      .WORD      .S.
025448      MOVE      SPAD <4>,MLR      ;

```

KMC11 ALU TESTS

9468		000075	NICPC=NICPC+1	
9469	025454	070204	.WORD .S.	
9470	025456		SBR 25	; SCOPE THE DATA....
9471		000076	NICPC=NICPC+1	
9472	025458	100453	.WORD .S.	
9473	025460		65: MOVE SPAD <4>,MLR	; RESET DATA POINTER...
9474		000077	NICPC=NICPC+1	
9475	025460	070204	.WORD .S.	
9476	025462		MOVE MEM,SPAD <0>,MARINC	; GET FIRST OPRAND...
9477		000100	NICPC=NICPC+1	
9478	025462	057220	.WORD .S.	
9479	025464		MOVE # 377,BREG	;
9480		000101	NICPC=NICPC+1	
9481	025464	000777	.WORD .S.	
9482	025466		SADD SPAD <2>,BREG	; SET C BIT...
9483	025470	000102	NICPC=NICPC+1	
9484	025470	060402	.WORD .S. ! DO	
9485		000103	LOAN MEM,SPAD <0>,BR.SP,MARINC	; SP 0 & BR = A OR NOTB
9486	025470	057640	NICPC=NICPC+1	
9487	025472		.WORD .S.	
9488		000104	MOVE MEM,SPAD <3>,MARINC	; DUMMY INSTR, TO MARINC.
9489	025472	057223	NICPC=NICPC+1	
9490	025474		.WORD .S.	
9491			SIFEQ MEM,SPAD <0> 9S	; BR IF GOOD...
9492	025474			
9493		000105	SUBC SPAD <0>,MEM,NOP	
9494	025474	040360	NICPC=NICPC+1	
9495	025476		.WORD .S.	
9496		000106	BZ 9S	
9497	025476	101521	NICPC=NICPC+1	
9498	025500		.WORD .S.	
9499		000107	MOVE MEM,OUT1 <CSR4>	; GOOD DATA.
9500	025500	041224	NICPC=NICPC+1	
9501	025502		.WORD .S.	
9502		000110	MOVE BREG,OUT1 <CSR5>	; BAD DATA.
9503	025504	061225	NICPC=NICPC+1	
9504		000111	.WORD .S.	
9505	025504	000423	MOVE # 23,BREG	; ERROR TYPE...
9506	025506		NICPC=NICPC+1	
9507		000112	MOVE BREG,OUT1 <CSR3>	;
9508	025506	061223	NICPC=NICPC+1	
9509	025510		.WORD .S.	
9510		000113	MOVE # 12,BREG	; ALU FUNCTION CODE.
9511	025510	000412	NICPC=NICPC+1	
9512	025512		.WORD .S.	
9513		000114	MOVE BREG,OUT1 <CSR7>	; LOAD IT...
9514	025512	061227	NICPC=NICPC+1	
9515	025514		.WORD .S.	
9516		000115	CALL EROR	; REPORT ERROR...
9517	025514	000517	MOVE # <NICPC+3>,BREG	
9518			NICPC=NICPC+1	
9519	025514		.WORD .S.	
9520	025516		SBR EROR	
9521		000116	NICPC=NICPC+1	

9571 025516 104400
9572 025520
9573 025520 000117
9574 025520 070204
9575 025520
9576 025522 000120
9577 025522 100477
9578 025524
9579 025524 000121
9580 025524 000523
9581 025526
9582 025526 000122
9583 025526 104427
9584 025530
9585 025530 000123
9586 025530 070204
9587 025532
9588 025532 000124
9589 025532 100477
9590 025534
9591 025534 000125
9592 025534 000404
9593 025536
9594 025536 077004
9595 025540
9596 025540 000126
9597 025540 063167
9598 025542
9599 025542 000127
9600 025542 101532
9601 025544
9602 025544 000130
9603 025544 100453
9604 025546
9605 025546 45:
9606 025546
9607 025546 000131
9608 025546 000533
9609 025550
9610 025550 000132
9611 025550 104454
9612 025552
9613 025552 000133
9614 025552 100400
9615 025554
9616 025554
9617 025554
9618 025554
9619 025554
9620 025554
9621 025554
9622 025554
9623 025554
9624 025554
9625 025554
9626 025554
9627 025554
9628 025554
9629 025554
9630 025554
9631 025554
9632 025554
9633 025554
9634 025554
9635 025554
9636 025554
9637 025554
9638 025554
9639 025554
9640 025554
9641 025554
9642 025554
9643 025554
9644 025554
9645 025554
9646 025554
9647 025554
9648 025554
9649 025554
9650 025554
9651 025554
9652 025554
9653 025554
9654 025554
9655 025554
9656 025554
9657 025554
9658 025554
9659 025554
9660 025554
9661 025554
9662 025554
9663 025554
9664 025554
9665 025554
9666 025554
9667 025554
9668 025554
9669 025554
9670 025554
9671 025554
9672 025554
9673 025554
9674 025554
9675 025554
9676 025554
9677 025554
9678 025554
9679 025554

```
.WORD .S.  
MOVE SPAD <4>,MLR ;RESTORE DATA POINTER.  
NICPC=NICPC+1  
.WORD .S.  
SBR 6S ;LOOP ON ERROR...  
NICPC=NICPC+1  
95: .WORD .S.  
CALL SCPI ;SCOPE THE ERROR...  
MOVE # <NICPC+3>,BREG  
NICPC=NICPC+1  
.WORD .S.  
SBR SCPI  
NICPC=NICPC+1  
.WORD .S.  
MOVE SPAD <4>,MLR ;RESTORE DATA POINTER...  
NICPC=NICPC+1  
.WORD .S.  
SBR 6S ;SCOPE THE DATA...  
NICPC=NICPC+1  
.WORD .S.  
MOVE # 4,BREG ;UPDATE BACKGROUND POINTER.  
NICPC=NICPC+1  
.WORD .S.  
SADD BREG,SPAD <4>,MARINC ;ALSO DATA POINTER.  
.WORD .S.!MARINC!.DO  
SDEC SPAD <7> ;IS IT DONE??  
NICPC=NICPC+1  
.WORD .S.!DSP  
BZ 4S ;YES, SCOPE THE TEST.  
NICPC=NICPC+1  
.WORD .S.  
SBR 2S ;DO, THE NEXT.  
NICPC=NICPC+1  
45: .WORD .S.  
CALL SCPE ;SCOPE THE TEST...  
MOVE # <NICPC+3>,BREG  
NICPC=NICPC+1  
.WORD .S.  
SBR SCPE  
NICPC=NICPC+1  
.WORD .S.  
SBR 1S ;DO THE NEXT ITERATION...  
NICPC=NICPC+1  
SALUT1 0,<A AND B>,AND,0,0,0,0,0,0,-1,-1,125,125,0,0,0,0,252,252,<A AND B>,1,13  
SXZ
```

```
***** TEST 45 *****  
*ALU TEST  
*TEST OF ALU FUNCTION A AND B WITH C BIT CLEARED.  
*TEST OF ALU FUNCTION A AND B WITH C BIT SET.  
*ALU FUNCTION (A AND B)  
*LOAD MAIN MEMORY 16 WORDS OF DATA.  
*PERFORM THE FUNCTION, VERIFY THE RESULTS..
```

SXZ

```

9580 ;:*****
9581 ;
9582 025554 $TSTN ; TEST 45
9583 ;-----
9584 ;
9585 025554 012737 000045 001202 TST45: MOV #45,$TSTNM ; LOAD THE NO. OF THIS TEST
9586 025562 012737 026102 001442 MOV #TST46,NEXT ; POINT TO THE START OF NEXT TEST.
9587 ;
9588 025570 004737 035536 JSR PC,LDVRWT ;R1 CONTAINS BASE KMC11 ADDRESS
9589 025574 025610 MCT45 ;LOAD-VERIFY-WAIT.
9590 025576 104022 ERROR 22 ;TIME OUT ERROR...
9591 025600 012706 001200 MOV #STACK,SP ;RESET STACK.
9592 025604 000177 153632 JMP @NEXT ;GO TO NEXT TEST...
9593 ;
9594 025610 MCT45: 15: MOVE #0,MLR ;SET MAR+LO.
9595 000000 MICPC=MICPC+1
9596 025610 010000 .WORD .S.
9597 025612 MOVE #0,MPR ;SET MAR+HI.
9598 000001 MICPC=MICPC+1
9599 025612 004000 .WORD .S.
9600 025614 MOVE #0,BREG ;
9601 000002 MICPC=MICPC+1
9602 025614 000400 .WORD .S.
9603 025616 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS...
9604 000003 MICPC=MICPC+1
9605 025616 063236 .WORD .S.
9606 025620 MOVE BREG,SPAD <0> ;
9607 000004 MICPC=MICPC+1
9608 025620 063220 .WORD .S.
9609 025622 MOVE BREG,SPAD <1> ;
9610 000005 MICPC=MICPC+1
9611 025622 063221 .WORD .S.
9612 025624 MOVE BREG,SPAD <2> ;
9613 000006 MICPC=MICPC+1
9614 025624 063222 .WORD .S.
9615 025626 SDEC SPAD <2> ;
9616 000007 MICPC=MICPC+1
9617 025626 063162 .WORD .S.!.DSP
9618 000010 MOVE BREG,SPAD <4> ;
9619 025630 063124 MICPC=MICPC+1
9620 000011 .WORD .S.
9621 025632 MOVE #0,MEM MARINC ;LOAD THE DATA IN MEMORY.
9622 016400 MICPC=MICPC+1
9623 000012 .WORD .S.
9624 025634 MOVE #0,MEM MARINC ;LOAD THE DATA IN MEMORY
9625 016400 MICPC=MICPC+1
9626 000013 .WORD .S.
9627 025636 MOVE #0,MEM MARINC
9628 000013 MICPC=MICPC+1
9629 025636 016400 .WORD .S.
9630 025640 MOVE #0,MEM MARINC ;RESULT WITH C BIT SET.
9631 000014 MICPC=MICPC+1
9632 025640 016400 .WORD .S.
9633 025642 MOVE #-1,MEM MARINC ;LOAD THE DATA IN MEMORY.
9634 000015 MICPC=MICPC+1
9635 025642 016777 .WORD .S.

```

9636	025644		MOVE # 0, MEM MARINC ;LOAD THE DATA IN MEMORY.
9637		000016	MICPC=MICPC+1
9638	025644	016400	.WORD .S.
9639	025646		MOVE # 0, MEM MARINC
9640		000017	MICPC=MICPC+1
9641	025646	016400	.WORD .S.
9642	025650		MOVE # 0, MEM MARINC ;RESULT WITH C BIT SET.
9643		000020	MICPC=MICPC+1
9644	025650	016400	.WORD .S.
9645	025652		MOVE # 0, MEM MARINC ;LOAD THE DATA IN MEMORY.
9646		000021	MICPC=MICPC+1
9647	025652	016400	.WORD .S.
9648	025654		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.
9649		000022	MICPC=MICPC+1
9650	025654	016777	.WORD .S.
9651	025656		MOVE # 0, MEM MARINC
9652		000023	MICPC=MICPC+1
9653	025656	016400	.WORD .S.
9654	025660		MOVE # 0, MEM MARINC ;RESULT WITH C BIT SET.
9655		000024	MICPC=MICPC+1
9656	025660	016400	.WORD .S.
9657	025662		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY
9658		000025	MICPC=MICPC+1
9659	025662	016777	.WORD .S.
9660	025664		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.
9661		000026	MICPC=MICPC+1
9662	025664	016777	.WORD .S.
9663	025666		MOVE # -1, MEM MARINC
9664		000027	MICPC=MICPC+1
9665	025666	016777	.WORD .S.
9666	025670		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.
9667		000030	MICPC=MICPC+1
9668	025670	016777	.WORD .S.
9669	025672		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
9670		000031	MICPC=MICPC+1
9671	025672	016525	.WORD .S.
9672	025674		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
9673		000032	MICPC=MICPC+1
9674	025674	016525	.WORD .S.
9675	025676		MOVE # 125, MEM MARINC
9676		000033	MICPC=MICPC+1
9677	025676	016525	.WORD .S.
9678	025700		MOVE # 125, MEM MARINC ;RESULT WITH C BIT SET.
9679		000034	MICPC=MICPC+1
9680	025700	016525	.WORD .S.
9681	025702		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
9682		000035	MICPC=MICPC+1
9683	025702	016652	.WORD .S.
9684	025704		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
9685		000036	MICPC=MICPC+1
9686	025704	016525	.WORD .S.
9687	025706		MOVE # 0, MEM MARINC
9688		000037	MICPC=MICPC+1
9689	025706	016400	.WORD .S.
9690	025710		MOVE # 0, MEM MARINC ;RESULT WITH C BIT SET.
9691		000040	MICPC=MICPC+1

9692	025710	016400	.WORD .S.	
9693	025712		MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
9694		000041	MICPC=MICPC+1	
9695	025712	016525	.WORD .S.	
9696	025714		MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY
9697		000042	MICPC=MICPC+1	
9698	025714	016652	.WORD .S.	
9699	025716		MOVE # 0, MEM MARINC	
9700		000043	MICPC=MICPC+1	
9701	025716	016400	.WORD .S.	
9702	025720		MOVE # 0, MEM MARINC	;RESULT WITH C BIT SET.
9703		000044	MICPC=MICPC+1	
9704	025720	016400	.WORD .S.	
9705	025722		MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY.
9706		000045	MICPC=MICPC+1	
9707	025722	016652	.WORD .S.	
9708	025724		MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY.
9709		000046	MICPC=MICPC+1	
9710	025724	016652	.WORD .S.	
9711	025726		MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY.
9712		000047	MICPC=MICPC+1	
9713	025726	016652	.WORD .S.	
9714	025730		MOVE # 252, MEM MARINC	;RESULT WITH C BIT SET.
9715		000050	MICPC=MICPC+1	
9716	025730	016652	.WORD .S.	
9717	025732		MOVE # 7, SPAD <7>	;SET THE COUNT.
9718		000051	MICPC=MICPC+1	
9719	025732	003007	.WORD .S.	
9720				
9721	025734		MOVE # 0, MLR	;MAR←0.
9722		000052	MICPC=MICPC+1	
9723	025734	010000	.WORD .S.	
9724				
9725	025736		25: MOVE # 0, BREG	;
9726		000053	MICPC=MICPC+1	
9727	025736	000400	.WORD .S.	
9728	025740		SADD SPAD <1>, BREG	;CLEAR C BIT.
9729		000054	MICPC=MICPC+1	
9730	025740	060401	.WORD .S. ! DO	
9731	025742		MOVE MEM, SPAD <0> MARINC	;GET THE FIRST OPERAND.
9732		000055	MICPC=MICPC+1	
9733	025742	057220	.WORD .S.	
9734	025744		AND MEM, SPAD <0>, BR.SP, MARINC	;SPAD <0>:=SFUNC.
9735		000056	MICPC=MICPC+1	
9736	025744	057660	.WORD .S.	
9737	025746		SIFEQ MEM, SPAD <0> 3S	;BRANCH IF GOOD.
9738				
9739				
9740	025746		SUB2C SPAD <0>, MEM, NOP	
9741		000057	MICPC=MICPC+1	
9742	025746	040360	.WORD .S.	
9743	025750		BZ 3S	
9744		000060	MICPC=MICPC+1	
9745	025750	101473	.WORD .S.	
9746	025752		MOVE MEM, OUT1 <4>	;GOOD DATA
9747		000061	MICPC=MICPC+1	

9748	025752	041224	.WORD .S.	
9749	025754		MOVE BREG,OUT1 <5>	;BAD DATA.
9750		000062	MICPC=MICPC+1	
9751	025754	061225	.WORD .S.	
9752	025756		MOVE #15,BREG	;SET TYPE OF ERROR.
9753		000063	MICPC=MICPC+1	
9754	025756	000415	.WORD .S.	
9755	025760		MOVE BREG,OUT1 <3>	;SET TYPE OF ERROR.
9756		000064	MICPC=MICPC+1	
9757	025760	061223	.WORD .S.	
9758	025762		MOVE #13,BREG	;LOAD FUNCTION CODE...
9759		000065	MICPC=MICPC+1	
9760	025762	000413	.WORD .S.	
9761	025764		MOVE BREG,OUT1 <CSR7>	;LOAD IT...
9762		000066	MICPC=MICPC+1	
9763	025764	061227	.WORD .S.	
9764	025766		CALL EROR	;ALU A AND B ERROR...
9765	025766		MOVE # <MICPC+3>,BREG	
9766		000067	MICPC=MICPC+1	
9767	025766	000471	.WORD .S.	
9768	025770		SBR EROR	
9769		000070	MICPC=MICPC+1	
9770	025770	104400	.WORD .S.	
9771	025772		MOVE SPAD <4>,MLR	;RESET DATA POINTER...
9772		000071	MICPC=MICPC+1	
9773	025772	070204	.WORD .S.	
9774	025774		SBR 25	;LOOP ON ERROR...
9775		000072	MICPC=MICPC+1	
9776	025774	100453	.WORD .S.	
9777	025776		CALL SCP1	
9778	025776		MOVE # <MICPC+3>,BREG	
9779		000073	MICPC=MICPC+1	
9780	025776	000475	.WORD .S.	
9781	026000		SBR SCP1	
9782		000074	MICPC=MICPC+1	
9783	026000	104427	.WORD .S.	
9784	026002		MOVE SPAD <4>,MLR	;
9785		000075	MICPC=MICPC+1	
9786	026002	070204	.WORD .S.	
9787	026004		SBR 25	;SCOPE THE DATA....
9788		000076	MICPC=MICPC+1	
9789	026004	100453	.WORD .S.	
9790	026006		MOVE SPAD <4>,MLR	;RESET DATA POINTER...
9791		000077	MICPC=MICPC+1	
9792	026006	070204	.WORD .S.	
9793	026010		MOVE MEM,SPAD <0>,MARINC	;GET FIRST OPRAND...
9794		000100	MICPC=MICPC+1	
9795	026010	057220	.WORD .S.	
9796	026012		MOVE #377,BREG	;
9797		000101	MICPC=MICPC+1	
9798	026012	000777	.WORD .S.	
9799	026014		SADD SPAD <2>,BREG	;SET C BIT...
9800		000102	MICPC=MICPC+1	
9801	026014	060402	.WORD .S. ! DO	
9802	026016		AND MEM,SPAD <0>,BR.SP,MARINC	;SP 0 & BR = A AND B
9803		000103	MICPC=MICPC+1	

9804	026016	057660	WORD .S.	
9805	026020		MOVE MEM, SPAD <3>, MARINC	; DUMMY INSTR, TO MARINC.
9806		000104	MICPC=MICPC+1	
9807	026020	057223	WORD .S.	
9808	026022		SIFEQ MEM, SPAD <0> 9S	; BR IF GOOD...
9809				
9810				
9811	026022		SUB2C SPAD <0>, MEM, NOP	
9812		000105	MICPC=MICPC+1	
9813	026022	040360	WORD .S.	
9814	026024		BZ 9S	
9815		000106	MICPC=MICPC+1	
9816	026024	101521	WORD .S.	
9817	026026		MOVE MEM, OUT1 <CSR4>	; GOOD DATA.
9818		000107	MICPC=MICPC+1	
9819	026026	041224	WORD .S.	
9820	026030		MOVE BREG, OUT1 <CSR5>	; BAD DATA.
9821		000110	MICPC=MICPC+1	
9822	026030	061225	WORD .S.	
9823	026032		MOVE # 23, BREG	; ERROR TYPE...
9824		000111	MICPC=MICPC+1	
9825	026032	000423	WORD .S.	
9826	026034		MOVE BREG, OUT1 <CSR3>	;
9827		000112	MICPC=MICPC+1	
9828	026034	061223	WORD .S.	
9829	026036		MOVE # 13, BREG	; ALU FUNCTION CODE.
9830		000113	MICPC=MICPC+1	
9831	026036	000413	WORD .S.	
9832	026040		MOVE BREG, OUT1 <CSR7>	; LOAD IT...
9833		000114	MICPC=MICPC+1	
9834	026040	061227	WORD .S.	
9835	026042		CALL EROR	; REPORT ERROR...
9836	026042		MOVE # <MICPC+3>, BREG	
9837		000115	MICPC=MICPC+1	
9838	026042	000517	WORD .S.	
9839	026044		SBR EROR	
9840		000116	MICPC=MICPC+1	
9841	026044	104400	WORD .S.	
9842	026046		MOVE SPAD <4>, MLR	; RESTORE DATA POINTER.
9843		000117	MICPC=MICPC+1	
9844	026046	070204	WORD .S.	
9845	026050		SBR 6S	; LOOP ON ERROR...
9846		000120	MICPC=MICPC+1	
9847	026050	100477	WORD .S.	
9848	026052		CALL SCP1	; SCOPE THE ERROR...
9849	026052		MOVE # <MICPC+3>, BREG	
9850		000121	MICPC=MICPC+1	
9851	026052	000523	WORD .S.	
9852	026054		SBR SCP1	
9853		000122	MICPC=MICPC+1	
9854	026054	104427	WORD .S.	
9855	026056		MOVE SPAD <4>, MLR	; RESTORE DATA POINTER...
9856		000123	MICPC=MICPC+1	
9857	026056	070204	WORD .S.	
9858	026060		SBR 6S	; SCOPE THE DATA...
9859		000124	MICPC=MICPC+1	

KMC11 ALU TESTS

9860	026060	100477			WORD	S		
9861	026062				MOVE	#4	BREG	;UPDATE BACKGROUND POINTER.
9862		000125			MICPC=MICPC+1			
9863	026062	000404			WORD	S		
9864	026064				SADD	BREG	SPAD (4)	MARINC ;ALSO DATA POINTER.
9865	026064	077004			WORD	S	MARINC!..DO	
9866	026066				SDEC	SPAD (7)		;IS IT DONE??
9867		000126			MICPC=MICPC+1			
9868	026066	063167			WORD	S	!..DSP	
9869	026070				BZ	45		;YES, SCOPE THE TEST.
9870		000127			MICPC=MICPC+1			
9871	026070	101532			WORD	S		
9872	026072				SBR	25		;DO, THE NEXT.
9873		000130			MICPC=MICPC+1			
9874	026072	100453			WORD	S		
9875	026074				CALL	SCOPE		;SCOPE THE TEST...
9876	026074				MOVE	# (MICPC+3)	BREG	
9877		000131			MICPC=MICPC+1			
9878	026074	000533			WORD	S		
9879	026076				SBR	SCOPE		
9880		000132			MICPC=MICPC+1			
9881	026076	104454			WORD	S		
9882	026100				SBR	15		;DO THE NEXT ITERATION...
9883		000133			MICPC=MICPC+1			
9884	026100	100400			WORD	S		
9885	026102				SALUT1	0	<A OR B>	OR,0,0,-1,-1,-1,-1,-1,-1,125,125,-1,-1,-1,-1,252,252,<A OR B>,1,14
9886	026102				SXZ			
9887								
9888								
9889								
9890								
9891								
9892								
9893								
9894								
9895								
9896	026102				SXZ			
9897								
9898								
9899	026102				STSTN			
9900								
9901								
9902	026102	012737	000046	001202	TST46:	MOV	#46,STSTN	; LOAD THE NO. OF THIS TEST
9903	026110	012737	026430	001442		MOV	#TST47,NEXT	; POINT TO THE START OF NEXT TEST.
9904								;R1 CONTAINS BASE KMC11 ADDRESS
9905	026116	004737	035536			JSR	PC,LDVRMT	;LOAD-VERIFY-WAIT.
9906	026122	026136				MCT46		
9907	026124	104022				ERROR	22	;TIME OUT ERROR...
9908	026126	012706	001200			MOV	#STACK,SP	;RESET STACK...
9909	026132	000177	153304			JMP	@NEXT	;GO TO NEXT TEST...
9910	026136				MCT46:			
9911	026136				IS:	MOVE	# 0,MLR	;SET MAR+LO.
9912		000000				MICPC=MICPC+1		
9913	026136	010000				WORD	S	
9914	026140					MOVE	# 0,MPR	;SET MAR+HI.
		000001				MICPC=MICPC+1		

```

***** TEST 46 *****
*ALU TEST
*TEST OF ALU FUNCTION A OR B WITH C BIT CLEARED.
*TEST OF ALU FUNCTION A OR B WITH C BIT SET.
*ALU FUNCTION (A OR B)
*LOAD MAIN MEMORY 16 WORDS OF DATA.
*PERFORM THE FUNCTION, VERIFY THE RESULTS..

```

9916	026140	004000	.WORD .S.	
9917	026142	000002	MOVE # 0, BREG	
9918		000002	MICPC=MICPC+1	
9919	026142	000400	.WORD .S.	
9920	026144		MOVE BREG, SPAD <16>	;FOR RETURN ADDRESS...
9921		000003	MICPC=MICPC+1	
9922	026144	063236	.WORD .S.	
9923	026146		MOVE BREG, SPAD <0>	;
9924		000004	MICPC=MICPC+1	
9925	026146	063220	.WORD .S.	
9926	026150		MOVE BREG, SPAD <1>	;
9927		000005	MICPC=MICPC+1	
9928	026150	063221	.WORD .S.	
9929	026152		MOVE BREG, SPAD <2>	;
9930		000006	MICPC=MICPC+1	
9931	026152	063222	.WORD .S.	
9932	026154		SDEC SPAD <2>	;
9933		000007	MICPC=MICPC+1	
9934	026154	063162	.WORD .S.	
9935	026156		MOVE BREG, SPAD <4>	;
9936		000010	MICPC=MICPC+1	
9937	026156	063224	.WORD .S.	
9938	026160		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
9939		000011	MICPC=MICPC+1	
9940	026160	016400	.WORD .S.	
9941	026162		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY
9942		000012	MICPC=MICPC+1	
9943	026162		.WORD .S.	
9944	026164		MOVE # 0, MEM MARINC	
9945		000013	MICPC=MICPC+1	
9946	026164		.WORD .S.	
9947	026166		MOVE # 0, MEM MARINC	;RESULT WITH C BIT SET.
9948		000014	MICPC=MICPC+1	
9949	026166		.WORD .S.	
9950	026170		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
9951		000015	MICPC=MICPC+1	
9952	026170	016777	.WORD .S.	
9953	026172		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
9954		000016	MICPC=MICPC+1	
9955	026172	016400	.WORD .S.	
9956		000017	MOVE # -1, MEM MARINC	
9957		000017	MICPC=MICPC+1	
9958		000017	.WORD .S.	
9959		000020	MOVE # -1, MEM MARINC	;RESULT WITH C BIT SET.
9960		000020	MICPC=MICPC+1	
9961	026176	016777	.WORD .S.	
9962	026200		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
9963		000021	MICPC=MICPC+1	
9964	026200	016400	.WORD .S.	
9965	026202		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
9966		000022	MICPC=MICPC+1	
9967	026202	016777	.WORD .S.	
9968	026204		MOVE # -1, MEM MARINC	
9969		000023	MICPC=MICPC+1	
9970	026204	016777	.WORD .S.	
9971	026206		MOVE # -1, MEM MARINC	;RESULT WITH C BIT SET.

9972		000024	MICPC=MICPC+1	
9973	026206	016777	.WORD .S.	
9974	026210		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY	
9975		000025	MICPC=MICPC+1	
9976	026210	016777	.WORD .S.	
9977	026212		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.	
9978		000026	MICPC=MICPC+1	
9979	026212	016777	.WORD .S.	
9980	026214		MOVE # -1, MEM MARINC	
9981		000027	MICPC=MICPC+1	
9982	026214	016777	.WORD .S.	
9983	026216		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.	
9984		000030	MICPC=MICPC+1	
9985	026216	016777	.WORD .S.	
9986	026220		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
9987		000031	MICPC=MICPC+1	
9988	026220	016525	.WORD .S.	
9989	026222		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
9990		000032	MICPC=MICPC+1	
9991	026222	016525	.WORD .S.	
9992	026224		MOVE # 125, MEM MARINC	
9993		000033	MICPC=MICPC+1	
9994	026224	016525	.WORD .S.	
9995	026226		MOVE # 125, MEM MARINC ;RESULT WITH C BIT SET.	
9996		000034	MICPC=MICPC+1	
9997	026226	016525	.WORD .S.	
9998	026230		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
9999		000035	MICPC=MICPC+1	
10000	026230	016652	.WORD .S.	
10001	026232		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
10002		000036	MICPC=MICPC+1	
10003	026232	016525	.WORD .S.	
10004	026234		MOVE # -1, MEM MARINC	
10005		000037	MICPC=MICPC+1	
10006	026234	016777	.WORD .S.	
10007	026236		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.	
10008		000040	MICPC=MICPC+1	
10009	026236	016777	.WORD .S.	
10010	026240		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
10011		000041	MICPC=MICPC+1	
10012	026240	016525	.WORD .S.	
10013	026242		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY	
10014		000042	MICPC=MICPC+1	
10015	026242	016652	.WORD .S.	
10016	026244		MOVE # -1, MEM MARINC	
10017		000043	MICPC=MICPC+1	
10018	026244	016777	.WORD .S.	
10019	026246		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.	
10020		000044	MICPC=MICPC+1	
10021	026246	016777	.WORD .S.	
10022	026250		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
10023		000045	MICPC=MICPC+1	
10024	026250	016652	.WORD .S.	
10025	026252		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
10026		000046	MICPC=MICPC+1	
10027	026252	016652	.WORD .S.	

10028	026254		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
10029		000047	MICPC=MICPC+1
10030	026254	016652	.WORD .S.
10031	026256		MOVE # 252, MEM MARINC ;RESULT WITH C BIT SET.
10032		000050	MICPC=MICPC+1
10033	026256	016652	.WORD .S.
10034	026260		MOVE # 7, SPAD <7> ;SET THE COUNT.
10035		000051	MICPC=MICPC+1
10036	026260	003007	.WORD .S.
10037			
10038	026262		MOVE # 0, MLR ;MAR+0.
10039		000052	MICPC=MICPC+1
10040	026262	010000	.WORD .S.
10041			
10042	026264		2S: MOVE # 0, BREG ;
10043		000053	MICPC=MICPC+1
10044	026264	000400	.WORD .S.
10045	026266		SADD SPAD <1>, BREG ;CLEAR C BIT.
10046		000054	MICPC=MICPC+1
10047	026266	060401	.WORD .S. !. DO
10048	026270		MOVE MEM, SPAD <0> MARINC ;GET THE FIRST OPERAND.
10049		000055	MICPC=MICPC+1
10050	026270	057220	.WORD .S.
10051	026272		OR MEM, SPAD <0>, BR. SP, MARINC ;SPAD <0>:=SFUNC.
10052		000056	MICPC=MICPC+1
10053	026272	057700	.WORD .S.
10054	026274		SIFEQ MEM, SPAD <0> 3S ;BRANCH IF GOOD.
10055			
10056			
10057	026274		SUB2C SPAD <0>, MEM, NOP
10058		000057	MICPC=MICPC+1
10059	026274	040360	.WORD .S.
10060	026276		BZ 3S
10061		000060	MICPC=MICPC+1
10062	026276	101473	.WORD .S.
10063	026300		MOVE MEM, OUT1 <4> ;GOOD DATA
10064		000061	MICPC=MICPC+1
10065	026300	041224	.WORD .S.
10066	026302		MOVE BREG, OUT1 <5> ;BAD DATA.
10067		000062	MICPC=MICPC+1
10068	026302	061225	.WORD .S.
10069	026304		MOVE # 15, BREG ;SET TYPE OF ERROR.
10070		000063	MICPC=MICPC+1
10071	026304	000415	.WORD .S.
10072	026306		MOVE BREG, OUT1 <3> ;SET TYPE OF ERROR.
10073		000064	MICPC=MICPC+1
10074	026306	061223	.WORD .S.
10075	026310		MOVE # 14, BREG ;LOAD FUNCTION CODE...
10076		000065	MICPC=MICPC+1
10077	026310	000414	.WORD .S.
10078	026312		MOVE BREG, OUT1 <CSR7> ;LOAD IT...
10079		000066	MICPC=MICPC+1
10080	026312	061227	.WORD .S.
10081	026314		CALL EROR ;ALU A OR B ERROR...
10082	026314		MOVE # <MICPC+3>, BREG
10083		000067	MICPC=MICPC+1

10084	026314	000471	.WORD .S.
10085	026316		SBR ERROR
10086		000070	MICPC=MICPC+1
10087	026316	104400	.WORD .S.
10088	026320		MOVE SPAD <4>,MLR ;RESET DATA POINTER...
10089		000071	MICPC=MICPC+1
10090	026320	070204	.WORD .S.
10091	026322		SBR 25 ;LOOP ON ERROR...
10092		000072	MICPC=MICPC+1
10093	026322	100453	.WORD .S.
10094	026324		35: CALL SCP1
10095	026324		MOVE 8 <MICPC+3>,BREG
10096		000073	MICPC=MICPC+1
10097	026324	000475	.WORD .S.
10098	026326		SBR SCP1
10099		000074	MICPC=MICPC+1
10100	026326	104427	.WORD .S.
10101	026330		MOVE SPAD <4>,MLR ;
10102		000075	MICPC=MICPC+1
10103	026330	070204	.WORD .S.
10104	026332		SBR 25 ;SCOPE THE DATA....
10105		000076	MICPC=MICPC+1
10106	026332	100453	.WORD .S.
10107	026334		65: MOVE SPAD <4>,MLR ;RESET DATA POINTER...
10108		000077	MICPC=MICPC+1
10109	026334	070204	.WORD .S.
10110	026336		MOVE MEM,SPAD <0>,MARINC ;GET FIRST OPRAND...
10111		000100	MICPC=MICPC+1
10112	026336	057220	.WORD .S.
10113	026340		MOVE 8 377,BREG ;
10114		000101	MICPC=MICPC+1
10115	026340	000777	.WORD .S.
10116	026342		SADD SPAD <2>,BREG ;SET C BIT...
10117		000102	MICPC=MICPC+1
10118	026342	060402	.WORD .S. ! DO
10119	026344		OR MEM,SPAD <0>,BR.SP,MARINC ;SP 0 & BR = A OR B
10120		000103	MICPC=MICPC+1
10121	026344	057700	.WORD .S.
10122	026346		MOVE MEM,SPAD <3>,MARINC ;DUMMY INSTR, TO MAK.INC.
10123		000104	MICPC=MICPC+1
10124	026346	057223	.WORD .S.
10125	026350		SIFEQ MEM,SPAD <0> 95 ;BR IF GOOD...
10126			
10127			
10128	026350		SUB2C SPAD <0>,MEM,NOP
10129		000105	MICPC=MICPC+1
10130	026350	040360	.WORD .S.
10131	026352		BZ 95
10132		000106	MICPC=MICPC+1
10133	026352	101521	.WORD .S.
10134	026354		MOVE MEM,OUT1 <CSR4> ;GOOD DATA.
10135		000107	MICPC=MICPC+1
10136	026354	041224	.WORD .S.
10137	026356		MOVE BREG,OUT1 <CSR5> ;BAD DATA.
10138		000110	MICPC=MICPC+1
10139	026356	061225	.WORD .S.

10140	026360		MOVE # 23, BREG	;ERROR TYPE...
10141		000111	MICPC=MICPC+1	
10142	026360	000423	.WORD .S.	
10143	026362		MOVE BREG, OUT1 (CSR3)	;
10144		000112	MICPC=MICPC+1	
10145	026362	061223	.WORD .S.	
10146	026364		MOVE # 14, BREG	;ALU FUNCTION CODE.
10147		000113	MICPC=MICPC+1	
10148	026364	000414	.WORD .S.	
10149	026366		MOVE BREG, OUT1 (CSR7)	;LOAD IT...
10150		000114	MICPC=MICPC+1	
10151	026366	061227	.WORD .S.	
10152	026370		CALL EROR ;REPORT ERROR...	
10153	026370		MOVE # (MICPC+3), BREG	
10154		000115	MICPC=MICPC+1	
10155	026370	000517	.WORD .S.	
10156	026372		SBR EROR	
10157		000116	MICPC=MICPC+1	
10158	026372	104400	.WORD .S.	
10159	026374		MOVE SPAD (4), MLR	;RESTORE DATA POINTER.
10160		000117	MICPC=MICPC+1	
10161	026374	070204	.WORD .S.	
10162	026376		SBR 6S	;LOOP ON ERROR...
10163		000120	MICPC=MICPC+1	
10164	026376	100477	.WORD .S.	
10165	026400		CALL SCP1 ;SCOPE THE ERROR...	
10166	026400		MOVE # (MICPC+3), BREG	
10167		000121	MICPC=MICPC+1	
10168	026400	000523	.WORD .S.	
10169	026402		SBR SCP1	
10170		000122	MICPC=MICPC+1	
10171	026402	104427	.WORD .S.	
10172	026404		MOVE SPAD (4), MLR	;RESTORE DATA POINTER...
10173		000123	MICPC=MICPC+1	
10174	026404	070204	.WORD .S.	
10175	026406		SBR 6S	;SCOPE THE DATA...
10176		000124	MICPC=MICPC+1	
10177	026406	100477	.WORD .S.	
10178	026410		MOVE # 4, BREG	;UPDATE BACKGROUND POINTER.
10179		000125	MICPC=MICPC+1	
10180	026410	000404	.WORD .S.	
10181	026412		SADD BREG, SPAD (4), MARINC	;ALSO DATA POINTER.
10182	026412	077004	.WORD .S. !MARINC!.DO	
10183	026414		SDEC SPAD (7)	;IS IT DONE??
10184		000126	MICPC=MICPC+1	
10185	026414	063167	.WORD .S. !.DSP	
10186	026416		BZ 4S	;YES, SCOPE THE TEST.
10187		000127	MICPC=MICPC+1	
10188	026416	101532	.WORD .S.	
10189	026420		SBR 2S	,M), THE NEXT.
10190		000130	MICPC=MICPC+1	
10191	026420	100453	.WORD .S.	
10192	026422		CALL SCPE ;SCOPE THE TEST...	
10193	026422		MOVE # (MICPC+3), BREG	
10194		000131	MICPC=MICPC+1	
10195	026422	000533	.WORD .S.	

```

10196 026424          SBR SCPE
10197          MICPC=MICPC+1
10198 026424 000132  .WORD .S.
10199 026426 104454  SBR 15 ;DO THE NEXT ITERATION...
10200          MICPC=MICPC+1
10201 026426 000133  .WORD .S.
10202 026430 100400  SALUT1 0,<A XOR B>,SXOR,0,0,-1,-1,-1,-1,0,0,0,0,-1,-1,-1,-1,0,0,<A XOR B>,1,15
10203 026430  SXZ
10204
10205
10206          ;***** TEST 47 *****
10207          ;*ALU TEST
10208          ;*TEST OF ALU FUNCTION A XOR B WITH C BIT CLEARED.
10209          ;*TEST OF ALU FUNCTION A XOR B WITH C BIT SET.
10210          ;*ALU FUNCTION (A XOR B)
10211          ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
10212          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
10213 026430  SXZ
10214
10215          ;*****
10216 026430  STSTM
10217
10218          ; TEST 47
10219          ;-----
10219 026430 012737 000047 001202 TST47: MOV #47,STSTM ; LOAD THE NO. OF THIS TEST
10220 026436 012737 026756 001442  MOV #TST50,NEXT ; POINT TO THE START OF NEXT TEST.
10221          ;R1 CONTAINS BASE KMC11 ADDRESS
10222 026444 004737 035536  JSR PC,LDRMT ;LOAD-VERIFY-WAIT.
10223 026450 026464  MCT47
10224 026452 104022  ERROR 22 ;TIME OUT ERROR...
10225 026454 012706 001200  MOV #STACK,SP ;RESET STACK...
10226 026460 000177 152756  JMP @NEXT ;GO TO NEXT TEST...
10227 026464
10228 026464  MCT47:
10229          15: MOVE #0,MLR ;SET MAR+LO.
10230          MICPC=MICPC+1
10231 026464 010000  .WORD .S.
10232 026466          MOVE #0,MPR ;SET MAR+HI.
10233 026466 000001  MICPC=MICPC+1
10234 026470 004000  .WORD .S.
10235          MOVE #0,BREG ;
10236 026470 000400  MICPC=MICPC+1
10237 026472          .WORD .S.
10238          MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS...
10239 026472 063236  MICPC=MICPC+1
10240 026474          .WORD .S.
10241          MOVE BREG,SPAD <0> ;
10242 026474 063220  MICPC=MICPC+1
10243 026476          .WORD .S.
10244          MOVE BREG,SPAD <1> ;
10245 026476 063221  MICPC=MICPC+1
10246 026500          .WORD .S.
10247          MOVE BREG,SPAD <2> ;
10248 026500 063222  MICPC=MICPC+1
10249 026502          .WORD .S.
10250          SDEC SPAD <2> ;
10251 026502 063162  MICPC=MICPC+1
          .WORD .S.!.DSP

```

10252	026504		MOVE BREG SPAD <4>	
10253		000010	MICPC=MICPC+1	
10254	026504	063224	.WORD .S.	
10255	026506		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
10256		000011	MICPC=MICPC+1	
10257	026506	016400	.WORD .S.	
10258	026510		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY
10259		000012	MICPC=MICPC+1	
10260	026510	016400	.WORD .S.	
10261	026512		MOVE # 0, MEM MARINC	
10262		000013	MICPC=MICPC+1	
10263	026512	016400	.WORD .S.	
10264	026514		MOVE # 0, MEM MARINC	;RESULT WITH C BIT SET.
10265		000014	MICPC=MICPC+1	
10266	026514	016400	.WORD .S.	
10267	026516		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
10268		000015	MICPC=MICPC+1	
10269	026516	016777	.WORD .S.	
10270	026520		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
10271		000016	MICPC=MICPC+1	
10272	026520	016400	.WORD .S.	
10273	026522		MOVE # -1, MEM MARINC	
10274		000017	MICPC=MICPC+1	
10275	026522	016777	.WORD .S.	
10276	026524		MOVE # -1, MEM MARINC	;RESULT WITH C BIT SET.
10277		000020	MICPC=MICPC+1	
10278	026524	016777	.WORD .S.	
10279	026526		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
10280		000021	MICPC=MICPC+1	
10281	026526	016400	.WORD .S.	
10282	026530		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
10283		000022	MICPC=MICPC+1	
10284	026530	016777	.WORD .S.	
10285	026532		MOVE # -1, MEM MARINC	
10286		000023	MICPC=MICPC+1	
10287	026532	016777	.WORD .S.	
10288	026534		MOVE # -1, MEM MARINC	;RESULT WITH C BIT SET.
10289		000024	MICPC=MICPC+1	
10290	026534	016777	.WORD .S.	
10291	026536		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY
10292		000025	MICPC=MICPC+1	
10293	026536	016777	.WORD .S.	
10294	026540		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
10295		000026	MICPC=MICPC+1	
10296	026540	016777	.WORD .S.	
10297	026542		MOVE # 0, MEM MARINC	
10298		000027	MICPC=MICPC+1	
10299	026542	016400	.WORD .S.	
10300	026544		MOVE # 0, MEM MARINC	;RESULT WITH C BIT SET.
10301		000030	MICPC=MICPC+1	
10302	026544	016400	.WORD .S.	
10303	026546		MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
10304		000031	MICPC=MICPC+1	
10305	026546	016525	.WORD .S.	
10306	026550		MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
10307		000032	MICPC=MICPC+1	

10308	026550	016525	.WORD .S.
10309	026552		MOVE # 0, MEM MARINC
10310		000033	MICPC=MICPC+1
10311	026552	016400	.WORD .S.
10312	026554		MOVE # 0, MEM MARINC ;RESULT WITH C BIT SET.
10313		000034	MICPC=MICPC+1
10314	026554	016400	.WORD .S.
10315	026556		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
10316		000035	MICPC=MICPC+1
10317	026556	016652	.WORD .S.
10318	026560		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
10319		000036	MICPC=MICPC+1
10320	026560	016525	.WORD .S.
10321	026562		MOVE # -1, MEM MARINC
10322		000037	MICPC=MICPC+1
10323	026562	016777	.WORD .S.
10324	026564		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.
10325		000040	MICPC=MICPC+1
10326	026564	016777	.WORD .S.
10327	026566		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
10328		000041	MICPC=MICPC+1
10329	026566	016525	.WORD .S.
10330	026570		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY
10331		000042	MICPC=MICPC+1
10332	026570	016652	.WORD .S.
10333	026572		MOVE # -1, MEM MARINC
10334		000043	MICPC=MICPC+1
10335	026572	016777	.WORD .S.
10336	026574		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.
10337		000044	MICPC=MICPC+1
10338	026574	016777	.WORD .S.
10339	026576		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
10340		000045	MICPC=MICPC+1
10341	026576	016652	.WORD .S.
10342	026600		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
10343		000046	MICPC=MICPC+1
10344	026600	016652	.WORD .S.
10345	026602		MOVE # 0, MEM MARINC ;LOAD THE DATA IN MEMORY.
10346		000047	MICPC=MICPC+1
10347	026602	016400	.WORD .S.
10348	026604		MOVE # 0, MEM MARINC ;RESULT WITH C BIT SET.
10349		000050	MICPC=MICPC+1
10350	026604	016400	.WORD .S.
10351	026606		MOVE # 7, SPAD <7> ;SET THE COUNT.
10352		000051	MICPC=MICPC+1
10353	026606	003007	.WORD .S.
10354			
10355	026610		MOVE # 0, MLR ;MAR+0.
10356		000052	MICPC=MICPC+1
10357	026610	010000	.WORD .S.
10358			
10359	026612		25: MOVE # 0, BREG ;
10360		000053	MICPC=MICPC+1
10361	026612	000400	.WORD .S.
10362	026614		SADD SPAD <1>, BREG ;CLEAR C BIT.
10363		000054	MICPC=MICPC+1

10364	026614	060401	.WORD .S. ! DO
10365	026616		MOVE MEM, SPAD <0> MARINC ;GET THE FIRST OPERAND.
10366		000055	MICPC=MICPC+1
10367	026616	057220	.WORD .S.
10368	026620		XOR MEM, SPAD <0>, BR.SP, MARINC ;SPAD <0>:=SFUNC.
10369		000056	MICPC=MICPC+1
10370	026620	057720	.WORD .S.
10371	026622		SIFEQ MEM, SPAD <0> 3\$;BRANCH IF GOOD.
10372			
10373			
10374	026622		SUB2C SPAD <0>, MEM, NOP
10375		000057	MICPC=MICPC+1
10376	026622	040360	.WORD .S.
10377	026624		BZ 3\$
10378		000060	MICPC=MICPC+1
10379	026624	101473	.WORD .S.
10380	026626		MOVE MEM, OUT1 <4> ;GOOD DATA
10381		000061	MICPC=MICPC+1
10382	026626	041224	.WORD .S.
10383	026630		MOVE BREG, OUT1 <5> ;BAD DATA.
10384		000062	MICPC=MICPC+1
10385	026630	061225	.WORD .S.
10386	026632		MOVE # 15 BREG ;SET TYPE OF ERROR.
10387		000063	MICPC=MICPC+1
10388	026632	000415	.WORD .S.
10389	026634		MOVE BREG, OUT1 <3> ;SET TYPE OF ERROR.
10390		000064	MICPC=MICPC+1
10391	026634	061223	.WORD .S.
10392	026636		MOVE # 15 BREG ;LOAD FUNCTION CODE...
10393		000065	MICPC=MICPC+1
10394	026636	000415	.WORD .S.
10395	026640		MOVE BREG, OUT1 <CSR7> ;LOAD IT...
10396		000066	MICPC=MICPC+1
10397	026640	061227	.WORD .S.
10398	026642		CALL EROR ;ALU A XOR B ERROR...
10399	026642		MOVE # <MICPC+3>, BREG
10400		000067	MICPC=MICPC+1
10401	026642	000471	.WORD .S.
10402	026644		SBR EROR
10403		000070	MICPC=MICPC+1
10404	026644	104400	.WORD .S.
10405	026646		MOVE SPAD <4>, MLR ;RESET DATA POINTER...
10406		000071	MICPC=MICPC+1
10407	026646	070204	.WORD .S.
10408	026650		SBR 2\$;LOOP ON ERROR...
10409		000072	MICPC=MICPC+1
10410	026650	100453	.WORD .S.
10411	026652		CALL SCP1
10412	026652		MOVE # <MICPC+3>, BREG
10413		000073	MICPC=MICPC+1
10414	026652	000475	.WORD .S.
10415	026654		SBR SCP1
10416		000074	MICPC=MICPC+1
10417	026654	104427	.WORD .S.
10418	026656		MOVE SPAD <4>, MLR ;
10419		000075	MICPC=MICPC+1

3\$:

10420	026656	070204	.WORD .S.	
10421	026660		SBR 23	; SCOPE THE DATA....
10422		000076	MICPC=MICPC+1	
10423	026660	100453	.WORD .S.	
10424	026662		65: MOVE SPAD <4>,MLR	; RESET DATA POINTER...
10425		000077	MICPC=MICPC+1	
10426	026662	070204	.WORD .S.	
10427	026664		MOVE MEM,SPAD <0>,MARINC	; GET FIRST OPRAND...
10428		000100	MICPC=MICPC+1	
10429	026664	057220	.WORD .S.	
10430	026666		MOVE # 377,BREG	;
10431		000101	MICPC=MICPC+1	
10432	026666	000777	.WORD .S.	
10433	026670		SADD SPAD <2>,BREG	; SET C BIT...
10434		000102	MICPC=MICPC+1	
10435	026670	060402	.WORD .S.! DO	
10436	026672		SXOR MEM,SPAD <0>,BR.SP,MARINC	; SP 0 & BR = A XOR B
10437		000103	MICPC=MICPC+1	
10438	026672	057720	.WORD .S.	
10439	026674		MOVE MEM,SPAD <3>,MARINC	; DUMMY INSTR, TO MARINC.
10440		000104	MICPC=MICPC+1	
10441	026674	057223	.WORD .S.	
10442	026676		SIFEQ MEM,SPAD <0> 95	; BR IF GOOD...
10443				
10444				
10445	026676		SUB2C SPAD <0>,MEM,NOP	
10446		000105	MICPC=MICPC+1	
10447	026676	040360	.WORD .S.	
10448	026700		BZ 95	
10449		000106	MICPC=MICPC+1	
10450	026700	101521	.WORD .S.	
10451	026702		MOVE MEM,OUT1 <CSR4>	; GOOD DATA.
10452		000107	MICPC=MICPC+1	
10453	026702	041224	.WORD .S.	
10454	026704		MOVE BREG,OUT1 <CSR5>	; BAD DATA.
10455		000110	MICPC=MICPC+1	
10456	026704	061225	.WORD .S.	
10457	026706		MOVE # 23,BREG	; ERROR TYPE...
10458		000111	MICPC=MICPC+1	
10459	026706	000423	.WORD .S.	
10460	026710		MOVE BREG,OUT1 <CSR3>	;
10461		000112	MICPC=MICPC+1	
10462	026710	061223	.WORD .S.	
10463	026712		MOVE # 15,BREG	; ALU FUNCTION CODE.
10464		000113	MICPC=MICPC+1	
10465	026712	000415	.WORD .S.	
10466	026714		MOVE BREG,OUT1 <CSR7>	; LOAD IT...
10467		000114	MICPC=MICPC+1	
10468	026714	061227	.WORD .S.	
10469	026716		CALL EROR	; REPORT ERROR...
10470	026716		MOVE # <MICPC+3>,BREG	
10471		000115	MICPC=MICPC+1	
10472	026716	000517	.WORD .S.	
10473	026720		SBR EROR	
10474		000116	MICPC=MICPC+1	
10475	026720	104400	.WORD .S.	

```

10476 026722 000117 MOVE SPAD (4),MLR ;RESTORE DATA POINTER.
10477 026722 070204 MICPC=MICPC+1
10478 026722 070204 .WORD .S.
10479 026724 .SBR 65 ;LOOP ON ERROR...
10480 000120 MICPC=MICPC+1
10481 026724 100477 .WORD .S.
10482 026726 95: CALL SCPI ;SCOPE THE ERROR...
10483 026726 MOVE # (MICPC+3),BREG
10484 000121 MICPC=MICPC+1
10485 026726 000523 .WORD .S.
10486 026730 .SBR SCPI
10487 000122 MICPC=MICPC+1
10488 026730 104427 .WORD .S.
10489 026732 MOVE SPAD (4),MLR ;RESTORE DATA POINTER...
10490 000123 MICPC=MICPC+1
10491 026732 070204 .WORD .S.
10492 026734 .SBR 65 ;SCOPE THE DATA...
10493 000124 MICPC=MICPC+1
10494 026734 100477 .WORD .S.
10495 026736 MOVE # 4,BREG ;UPDATE BACKGROUND POINTER.
10496 000125 MICPC=MICPC+1
10497 026736 000404 .WORD .S.
10498 026740 SADD BREG,SPAD (4),MARINC ;ALSO DATA POINTER.
10499 026740 .WORD .S. !MARINC!.DO
10500 026742 SDEC SPAD (7) ;IS IT DONE??
10501 000126 MICPC=MICPC+1
10502 026742 063167 .WORD .S.!.DSP
10503 026744 BZ 45 ;YES, SCOPE THE TEST.
10504 000127 MICPC=MICPC+1
10505 026744 101532 .WORD .S.
10506 026746 .SBR 25 ;DO, THE NEXT.
10507 000130 MICPC=MICPC+1
10508 026746 100453 .WORD .S.
10509 026750 45: CALL SCPE ;SCOPE THE TEST...
10510 026750 MOVE # (MICPC+3),BREG
10511 000131 MICPC=MICPC+1
10512 026750 000533 .WORD .S.
10513 026752 .SBR SCPE
10514 000132 MICPC=MICPC+1
10515 026752 104454 .WORD .S.
10516 026754 .SBR 15 ;DO THE NEXT ITERATION...
10517 000133 MICPC=MICPC+1
10518 026754 100400 .WORD .S.
10519 026756 SALUT1 0,(ADD),SADD,0,0,-1,-1,-1,-1,376,376,252,252,-1,-1,-1,-1,124,124,(A PLUS B),1,00
10520 026756 SXZ
10521
10522
10523
10524
10525
10526
10527
10528
10529
10530 026756 SXZ
10531

```

;***** TEST 50 *****
;#ALU TEST
;#TEST OF ALU FUNCTION ADD WITH C BIT CLEARED.
;#TEST OF ALU FUNCTION ADD WITH C BIT SET.
;#ALU FUNCTION (A PLUS B)
;#LOAD MAIN MEMORY 16 WORDS OF DATA.
;#PERFORM THE FUNCTION, VERIFY THE RESULTS..
;*****

```

10532
10533 026756          STSTN
10534                ; TEST 50
10535                ; -----
10536 026756 012737 000050 001202 TST50:  MOV     #50,STSTNM          ; LOAD THE NO. OF THIS TEST
10537 026764 012737 027304 001442      MOV     #TST51,NEXT        ; POINT TO THE START OF NEXT TEST.
10538                ; R1 CONTAINS BASE KMC11 ADDRESS
10539 026772 004737 035536          JSR     PC,LDRVMT         ; LOAD-VERIFY-WAIT.
10540 026776 027012          MCT50
10541 027000 104022          ERROR    22              ; TIME OUT ERROR...
10542 027002 012706 001200      MOV     #STACK,SP        ; RESET STACK...
10543 027006 000177 152430      JMP     @NEXT            ; GO TO NEXT TEST...
10544 027012          MCT50:
10545 027012          15:  MOVE     # 0,MLR          ; SET MAR+LO.
10546                MICPC=MICPC+1
10547 027012 010000          .WORD   .S.
10548 027014          MOVE     # 0,NPR          ; SET MAR+HI.
10549                MICPC=MICPC+1
10550 027014 004000          .WORD   .S.
10551 027016          MOVE     # 0,BREG
10552                MICPC=MICPC+1
10553 027016 000400          .WORD   .S.
10554 027020          MOVE     BREG,SPAD <16> ; FOR RETURN ADDRESS...
10555                MICPC=MICPC+1
10556 027020 063236          .WORD   .S.
10557 027022          MOVE     BREG,SPAD <0> ;
10558                MICPC=MICPC+1
10559 027022 063220          .WORD   .S.
10560 027024          MOVE     BREG,SPAD <1> ;
10561                MICPC=MICPC+1
10562 027024 063221          .WORD   .S.
10563 027026          MOVE     BREG,SPAD <2> ;
10564                MICPC=MICPC+1
10565 027026 063222          .WORD   .S.
10566 027030          SDEC    SPAD <2> ;
10567                MICPC=MICPC+1
10568 027030 063162          .WORD   .S. ; DSP
10569 027032          MOVE     BREG,SPAD <4> ;
10570                MICPC=MICPC+1
10571 027032 063224          .WORD   .S.
10572 027034          MOVE     # 0, MEM MARINC ; LOAD THE DATA IN MEMORY.
10573                MICPC=MICPC+1
10574 027034 016400          .WORD   .S.
10575 027036          MOVE     # 0, MEM MARINC ; LOAD THE DATA IN MEMORY
10576                MICPC=MICPC+1
10577 027036 016400          .WORD   .S.
10578 027040          MOVE     # 0, MEM MARINC
10579                MICPC=MICPC+1
10580 027040 016400          .WORD   .S.
10581 027042          MOVE     # 0, MEM MARINC ; RESULT WITH C BIT SET.
10582                MICPC=MICPC+1
10583 027042 016400          .WORD   .S.
10584 027044          MOVE     # -1, MEM MARINC ; LOAD THE DATA IN MEMORY.
10585                MICPC=MICPC+1
10586 027044 016777          .WORD   .S.
10587 027046          MOVE     # 0, MEM MARINC ; LOAD THE DATA IN MEMORY.

```

10588		000016	NICPC=NICPC+1
10589	027046	016400	.WORD .S.
10590	027050		MOVE # -1, MEM MARINC
10591		000017	NICPC=NICPC+1
10592	027050	016777	.WORD .S.
10593	027052		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.
10594		000020	NICPC=NICPC+1
10595	027052	016777	.WORD .S.
10596	027054		MOVE # 0, MEM MARINC ;LOAD THE DATA IN MEMORY.
10597		000021	NICPC=NICPC+1
10598	027054	016400	.WORD .S.
10599	027056		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.
10600		000022	NICPC=NICPC+1
10601	027056	016777	.WORD .S.
10602	027060		MOVE # -1, MEM MARINC
10603		000023	NICPC=NICPC+1
10604	027060	016777	.WORD .S.
10605	027062		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.
10606		000024	NICPC=NICPC+1
10607	027062	016777	.WORD .S.
10608	027064		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY
10609		000025	NICPC=NICPC+1
10610	027064	016777	.WORD .S.
10611	027066		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.
10612		000026	NICPC=NICPC+1
10613	027066	016777	.WORD .S.
10614	027070		MOVE # 376, MEM MARINC
10615		000027	NICPC=NICPC+1
10616	027070	016776	.WORD .S.
10617	027072		MOVE # 376, MEM MARINC ;RESULT WITH C BIT SET.
10618		000030	NICPC=NICPC+1
10619	027072	016776	.WORD .S.
10620	027074		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
10621		000031	NICPC=NICPC+1
10622	027074	016525	.WORD .S.
10623	027076		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
10624		000032	NICPC=NICPC+1
10625	027076	016525	.WORD .S.
10626	027100		MOVE # 252, MEM MARINC
10627		000033	NICPC=NICPC+1
10628	027100	016652	.WORD .S.
10629	027102		MOVE # 252, MEM MARINC ;RESULT WITH C BIT SET.
10630		000034	NICPC=NICPC+1
10631	027102	016652	.WORD .S.
10632	027104		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
10633		000035	NICPC=NICPC+1
10634	027104	016652	.WORD .S.
10635	027106		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
10636		000036	NICPC=NICPC+1
10637	027106	016525	.WORD .S.
10638	027110		MOVE # -1, MEM MARINC
10639		000037	NICPC=NICPC+1
10640	027110	016777	.WORD .S.
10641	027112		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.
10642		000040	NICPC=NICPC+1
10643	027112	016777	.WORD .S.

10644	027114		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
10645		000041	MICPC=MICPC+1
10646	027114	016525	.WORD .S.
10647	027116		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY
10648		000042	MICPC=MICPC+1
10649	027116	016652	.WORD .S.
10650	027120		MOVE # -1, MEM MARINC
10651		000043	MICPC=MICPC+1
10652	027120	016777	.WORD .S.
10653	027122		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.
10654		000044	MICPC=MICPC+1
10655	027122	016777	.WORD .S.
10656	027124		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
10657		000045	MICPC=MICPC+1
10658	027124	016652	.WORD .S.
10659	027126		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
10660		000046	MICPC=MICPC+1
10661	027126	016652	.WORD .S.
10662	027130		MOVE # 124, MEM MARINC ;LOAD THE DATA IN MEMORY.
10663		000047	MICPC=MICPC+1
10664	027130	016524	.WORD .S.
10665	027132		MOVE # 124, MEM MARINC ;RESULT WITH C BIT SET.
10666		000050	MICPC=MICPC+1
10667	027132	016524	.WORD .S.
10668	027134		MOVE # 7, SPAD <7> ;SET THE COUNT.
10669		000051	MICPC=MICPC+1
10670	027134	003007	.WORD .S.
10671			
10672	027136		MOVE # 0, MLR ;MAR←0.
10673		000052	MICPC=MICPC+1
10674	027136	010000	.WORD .S.
10675			
10676	027140		2S: MOVE # 0, BREG ;
10677		000053	MICPC=MICPC+1
10678	027140	000400	.WORD .S.
10679	027142		SADD SPAD <1>, BREG ;CLEAR C BIT.
10680		000054	MICPC=MICPC+1
10681	027142	060401	.WORD .S. 00
10682	027144		MOVE MEM, SPAD <0> MARINC ;GET THE FIRST OPERAND.
10683		000055	MICPC=MICPC+1
10684	027144	057220	.WORD .S.
10685	027146		SADD MEM, SPAD <0>, BR.SP, MARINC ;SPAD <0>:=SFUNC.
10686		000056	MICPC=MICPC+1
10687	027146	057400	.WORD .S.
10688	027150		SIFEQ MEM, SPAD <0> 3S ;BRANCH IF GOOD.
10689			
10690			
10691	027150		SUB2C SPAD <0>, MEM, NOP
10692		000057	MICPC=MICPC+1
10693	027150	040360	.WORD .S.
10694	027152		BZ 3S
10695		000060	MICPC=MICPC+1
10696	027152	101473	.WORD .S.
10697	027154		MOVE MEM, OUT1 <4> ;GOOD DATA
10698		000061	MICPC=MICPC+1
10699	027154	041224	.WORD .S.

10700	027156		MOVE BREG OUT1 <5>	;BAD DATA.
10701		000062	MICPC=MICPC+1	
10702	027156	061225	.WORD .S.	
10703	027160		MOVE #15,BREG	;SET TYPE OF ERROR.
10704		000063	MICPC=MICPC+1	
10705	027160	000415	.WORD .S.	
10706	027162		MOVE BREG OUT1 <3>	;SET TYPE OF ERROR.
10707		000064	MICPC=MICPC+1	
10708	027162	061223	.WORD .S.	
10709	027164		MOVE #00,BREG	;LOAD FUNCTION CODE...
10710		000065	MICPC=MICPC+1	
10711	027164	000400	.WORD .S.	
10712	027166		MOVE BREG OUT1 <CSR7>	;LOAD IT...
10713		000066	MICPC=MICPC+1	
10714	027166	061227	.WORD .S.	
10715	027170		CALL EROR	;ALU ADD ERROR...
10716	027170		MOVE # <MICPC+3>,BREG	
10717		000067	MICPC=MICPC+1	
10718	027170	000471	.WORD .S.	
10719	027172		SBR EROR	
10720		000070	MICPC=MICPC+1	
10721	027172	104400	.WORD .S.	
10722	027174		MOVE SPAD <4>,MLR	;RESET DATA POINTER...
10723		000071	MICPC=MICPC+1	
10724	027174	070204	.WORD .S.	
10725	027176		SBR 25	;LOOP ON ERROR...
10726		000072	MICPC=MICPC+1	
10727	027176	100453	.WORD .S.	
10728	027200		CALL SCP1	
10729	027200		MOVE # <MICPC+3>,BREG	
10730		000073	MICPC=MICPC+1	
10731	027200	000475	.WORD .S.	
10732	027202		SBR SCP1	
10733		000074	MICPC=MICPC+1	
10734	027202	104427	.WORD .S.	
10735	027204		MOVE SPAD <4>,MLR	;
10736		000075	MICPC=MICPC+1	
10737	027204	070204	.WORD .S.	
10738	027206		SBR 25	;SCOPE THE DATA....
10739		000076	MICPC=MICPC+1	
10740	027206	100453	.WORD .S.	
10741	027210		MOVE SPAD <4>,MLR	;RESET DATA POINTER...
10742		000077	MICPC=MICPC+1	
10743	027210	070204	.WORD .S.	
10744	027212		MOVE MEM,SPAD <0>,MARINC	;GET FIRST OPRAND...
10745		000100	MICPC=MICPC+1	
10746	027212	057220	.WORD .S.	
10747	027214		MOVE #377,BREG	;
10748		000101	MICPC=MICPC+1	
10749	027214	000777	.WORD .S.	
10750	027216		SADD SPAD <2>,BREG	;SET C BIT...
10751		000102	MICPC=MICPC+1	
10752	027216	060402	.WORD .S.!.DO	
10753	027220		SADD MEM,SPAD <0>,BR.SP,MARINC	;SP 0 & BR = ADD
10754		000103	MICPC=MICPC+1	
10755	027220	057400	.WORD .S.	


```

10756 027222      MOVE    MEM,SPAD <3>,MARINC    ;DUMMY INSTR, TO MARINC.
10757          MICPC=MICPC+1
10758 027222 000104      .WORD   .S.
10759 027224 057223      SIFEQ   MEM,SPAD <0>    9S    ;BR IF GOOD...
10760
10761
10762 027224      SUB2C   SPAD <0>,MEM,NOP
10763          MICPC=MICPC+1
10764 027224 040360      .WORD   .S.
10765 027226      BZ      9S
10766          MICPC=MICPC+1
10767 027226 101521      .WORD   .S.
10768 027230      MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
10769          MICPC=MICPC+1
10770 027230 041224      .WORD   .S.
10771 027232      MOVE    BREG,OUT1 <CSR5>    ;BAD DATA.
10772          MICPC=MICPC+1
10773 027232 061225      .WORD   .S.
10774 027234      MOVE    # 23,BREG          ;ERROR TYPE...
10775          MICPC=MICPC+1
10776 027234 000423      .WORD   .S.
10777 027236      MOVE    BREG,OUT1 <CSR3>    ;
10778          MICPC=MICPC+1
10779 027236 061223      .WORD   .S.
10780 027240      MOVE    # 00,BREG          ;ALU FUNCTION CODE.
10781          MICPC=MICPC+1
10782 027240 000400      .WORD   .S.
10783 027242      MOVE    BREG,OUT1 <CSR7>    ;LOAD IT...
10784          MICPC=MICPC+1
10785 027242 061227      .WORD   .S.
10786 027244      CALL    EROR ;REPORT ERROR...
10787 027244      MOVE    # <MICPC+3>,BREG
10788          MICPC=MICPC+1
10789 027244 000517      .WORD   .S.
10790 027246      SBR    EROR
10791          MICPC=MICPC+1
10792 027246 104400      .WORD   .S.
10793 027250      MOVE    SPAD <4>,MLR          ;RESTORE DATA POINTER.
10794          MICPC=MICPC+1
10795 027250 070204      .WORD   .S.
10796 027252      SBR    6S          ;LOOP ON ERROR...
10797          MICPC=MICPC+1
10798 027252 100477      .WORD   .S.
10799 027254      CALL    SCP1 ;SCOPE THE ERROR...
10800 027254 000121      MOVE    # <MICPC+3>,BREG
10801          MICPC=MICPC+1
10802 027254 000523      .WORD   .S.
10803 027256      SBR    SCP1
10804          MICPC=MICPC+1
10805 027256 104427      .WORD   .S.
10806 027260      MOVE    SPAD <4>,MLR          ;RESTORE DATA POINTER...
10807          MICPC=MICPC+1
10808 027260 070204      .WORD   .S.
10809 027262      SBR    6S          ;SCOPE THE DATA...
10810          MICPC=MICPC+1
10811 027262 100477      .WORD   .S.

```

9S:

```

10812 027264          MOVE      # 4, BREG          ;UPDATE BACKGROUND POINTER.
10813          MICPC=MICPC+1
10814 027264 000125   .WORD      .S.
10815 027266          SADD      BREG, SPAD (4), MARINC    ;ALSO DATA POINTER.
10816 027266 077004   .WORD      .S. !MARINC!..DO
10817 027270          SDEC      SPAD (7)          ;IS IT DONE??
10818          MICPC=MICPC+1
10819 027270 063167   .WORD      .S.!.DSP
10820 027272          BZ          45          ;YES, SCOPE THE TEST.
10821          MICPC=MICPC+1
10822 027272 101532   .WORD      .S.
10823 027274          SBR          25          ;DO, THE NEXT.
10824          MICPC=MICPC+1
10825 027274 000130   .WORD      .S.
10826 027276 100453   .WORD      .S.
10827 027276 45:     CALL      SCOPE          ;SCOPE THE TEST...
10828          MOVE      # (MICPC+3), BREG
10829 027276 000131   MICPC=MICPC+1
10830 027300 000533   .WORD      .S.
10831          SBR          SCOPE
10832 027300 000132   MICPC=MICPC+1
10833 027302 104454   .WORD      .S.
10834          SBR          15          ;DO THE NEXT ITERATION...
10835 027302 000133   MICPC=MICPC+1
10836 027304 100400   .WORD      .S.
10837 027304 SALUT: 0, <2A W/C>, SROL, 0, 1, 376, -1, 0, 1, 376, -1, 252, 253, 124, 125, 252, 253, 124, 125, <A PLUS A
SXZ
10838
10839
10840
10841          ;***** TEST 51 *****
10842          ;*ALU TEST
10843          ;*TEST OF ALU FUNCTION 2A W/C WITH C BIT CLEARED.
10844          ;*TEST OF ALU FUNCTION 2A W/C WITH C BIT SET.
10845          ;*ALU FUNCTION (A PLUS A PLUS C)
10846          ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
10847 027304 45:     SXZ
10848          ;*****
10849
10850 027304 STSTN
10851          ; TEST 51
10852          ;-----
10853 027304 012737 000051 001202 TST51: MOV      #51, STSTNM          ; LOAD THE NO. OF THIS TEST
10854 027312 012737 027632 001442 MOV      #TST52, NEXT          ; POINT TO THE START OF NEXT TEST.
10855          ;R1 CONTAINS BASE KMC11 ADDRESS
10856 027320 004737 035536 JSR      PC, LDVWRT          ;LOAD-VERIFY-WAIT.
10857 027324 027340 MCT51: MCT51
10858 027326 104022 ERROR    22
10859 027330 012706 001200 MOV      #STACK, SP          ; TIME OUT ERROR...
10860 027334 000177 152102 JMP      @NEXT          ; RESET STACK...
10861 027340 MCT51: ; GO TO NEXT TEST...
10862 027340 15:     MOVE      # 0, MLR          ;SET MAR+LO.
10863          MICPC=MICPC+1
10864 027340 010000   .WORD      .S.
10865 027342          MOVE      # 0, MPR          ;SET MAR+HI.
10866          MICPC=MICPC+1
10867 027342 004000   .WORD      .S.

```

10868	027344		MOVE	# 0, BREG	;
10869		000002	MICPC=MICPC+1		
10870	027344	000400	.WORD	.S.	
10871	027346		MOVE	BREG SPAD <16>	;FOR RETURN ADDRESS...
10872		000003	MICPC=MICPC+1		
10873	027346	063236	.WORD	.S.	
10874	027350		MOVE	BREG SPAD <0>	;
10875		000004	MICPC=MICPC+1		
10876	027350	063220	.WORD	.S.	
10877	027352		MOVE	BREG SPAD <1>	;
10879		000005	MICPC=MICPC+1		
10879	027352	063221	.WORD	.S.	
10880	027354		MOVE	BREG SPAD <2>	;
10881		000006	MICPC=MICPC+1		
10882	027354	063222	.WORD	.S.	
10883	027356		SDEC	SPAD <2>	;
10884		000007	MICPC=MICPC+1		
10885	027356	063162	.WORD	.S. DSP	
10886	027360		MOVE	BREG SPAD <4>	;
10887		000010	MICPC=MICPC+1		
10888	027360	063224	.WORD	.S.	
10889	027362		MOVE	# 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
10890		000011	MICPC=MICPC+1		
10891	027362	016400	.WORD	.S.	
10892	027364		MOVE	# 0, MEM MARINC	;LOAD THE DATA IN MEMORY
10893		000012	MICPC=MICPC+1		
10894	027364	016400	.WORD	.S.	
10895	027366		MOVE	# 0, MEM MARINC	
10896		000013	MICPC=MICPC+1		
10897	027366	016400	.WORD	.S.	
10898	027370		MOVE	# 1, MEM MARINC	;RESULT WITH C BIT SET.
10899		000014	MICPC=MICPC+1		
10900	027370	016401	.WORD	.S.	
10901	027372		MOVE	# -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
10902		000015	MICPC=MICPC+1		
10903	027372	016777	.WORD	.S.	
10904	027374		MOVE	# 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
10905		000016	MICPC=MICPC+1		
10906	027374	016400	.WORD	.S.	
10907	027376		MOVE	# 376, MEM MARINC	
10908		000017	MICPC=MICPC+1		
10909	027376	016776	.WORD	.S.	
10910	027400		MOVE	# -1, MEM MARINC	;RESULT WITH C BIT SET.
10911		000020	MICPC=MICPC+1		
10912	027400	016777	.WORD	.S.	
10913	027402		MOVE	# 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
10914		000021	MICPC=MICPC+1		
10915	027402	016400	.WORD	.S.	
10916	027404		MOVE	# -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
10917		000022	MICPC=MICPC+1		
10918	027404	016777	.WORD	.S.	
10919	027406		MOVE	# 0, MEM MARINC	
10920		000023	MICPC=MICPC+1		
10921	027406	016400	.WORD	.S.	
10922	027410		MOVE	# 1, MEM MARINC	;RESULT WITH C BIT SET.
10923		000024	MICPC=MICPC+1		

10924	027410	016401	.WORD .S.	
10925	027412		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY	
10926		000025	MICPC=MICPC+1	
10927	027412	016777	.WORD .S.	
10928	027414		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.	
10929		000026	MICPC=MICPC+1	
10930	027414	016777	.WORD .S.	
10931	027416		MOVE # 376, MEM MARINC	
10932		000027	MICPC=MICPC+1	
10933	027416	016776	.WORD .S.	
10934	027420		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.	
10935		000030	MICPC=MICPC+1	
10936	027420	016777	.WORD .S.	
10937	027422		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
10938		000031	MICPC=MICPC+1	
10939	027422	016525	.WORD .S.	
10940	027424		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
10941		000032	MICPC=MICPC+1	
10942	027424	016525	.WORD .S.	
10943	027426		MOVE # 252, MEM MARINC	
10944		000033	MICPC=MICPC+1	
10945	027426	016652	.WORD .S.	
10946	027430		MOVE # 253, MEM MARINC ;RESULT WITH C BIT SET.	
10947		000034	MICPC=MICPC+1	
10948	027430	016653	.WORD .S.	
10949	027432		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
10950		000035	MICPC=MICPC+1	
10951	027432	016652	.WORD .S.	
10952	027434		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
10953		000036	MICPC=MICPC+1	
10954	027434	016525	.WORD .S.	
10955	027436		MOVE # 124, MEM MARINC	
10956		000037	MICPC=MICPC+1	
10957	027436	016524	.WORD .S.	
10958	027440		MOVE # 125, MEM MARINC ;RESULT WITH C BIT SET.	
10959		000040	MICPC=MICPC+1	
10960	027440	016525	.WORD .S.	
10961	027442		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
10962		000041	MICPC=MICPC+1	
10963	027442	016525	.WORD .S.	
10964	027444		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY	
10965		000042	MICPC=MICPC+1	
10966	027444	016652	.WORD .S.	
10967	027446		MOVE # 252, MEM MARINC	
10968		000043	MICPC=MICPC+1	
10969	027446	016652	.WORD .S.	
10970	027450		MOVE # 253, MEM MARINC ;RESULT WITH C BIT SET.	
10971		000044	MICPC=MICPC+1	
10972	027450	016653	.WORD .S.	
10973	027452		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
10974		000045	MICPC=MICPC+1	
10975	027452	016652	.WORD .S.	
10976	027454		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
10977		000046	MICPC=MICPC+1	
10978	027454	016652	.WORD .S.	
10979	027456		MOVE # 124, MEM MARINC ;LOAD THE DATA IN MEMORY.	

10980		000047	MICPC=MICPC+1
10981	027456	016524	.WORD .S.
10982	027460		MOVE # 125, MEM MARINC ; RESULT WITH C BIT SET.
10983		000050	MICPC=MICPC+1
10984	027460	016525	.WORD .S.
10985	027462		MOVE # 7, SPAD <7> ; SET THE COUNT.
10986		000051	MICPC=MICPC+1
10987	027462	003007	.WORD .S.
10988			
10989	027464		MOVE # 0, MLR ; MAR+0.
10990		000052	MICPC=MICPC+1
10991	027464	010000	.WORD .S.
10992			
10993	027466		25: MOVE # 0, BREG ;
10994		000053	MICPC=MICPC+1
10995	027456	000400	.WORD .S.
10996	027470		SADD SPAD <1>, BREG ; CLEAR C BIT.
10997		000054	MICPC=MICPC+1
10998	027470	760401	.WORD .S. ; DO
10999	027472		MOVE MEM, SPAD <0> MARINC ; GET THE FIRST OPERAND.
11000		000055	MICPC=MICPC+1
11001	027472	057220	.WORD .S.
11002	027474		SROL SPAD <0>, BR.SP, MARINC ;
11003		000056	MICPC=MICPC+1
11004	027474	077540	.WORD .S.
11005	027476		SIFEG MEM, SPAD <0> 3S ; BRANCH IF GOOD.
11006			
11007			
11008	027476		SUB2C SPAD <0>, MEM, NOP
11009		000057	MICPC=MICPC+1
11010	027476	040360	.WORD .S.
11011	027500		BZ 3S
11012		000060	MICPC=MICPC+1
11013	027500	101473	.WORD .S.
11014	027502		MOVE MEM, OUT1 <4> ; GOOD DATA
11015		000061	MICPC=MICPC+1
11016	027502	041224	.WORD .S.
11017	027504		MOVE BREG, OUT1 <5> ; BAD DATA.
11018		000062	MICPC=MICPC+1
11019	027504	061225	.WORD .S.
11020	027506		MOVE # 15, BREG ; SET TYPE OF ERROR.
11021		000063	MICPC=MICPC+1
11022	027506	000415	.WORD .S.
11023	027510		MOVE BREG, OUT1 <3> ; SET TYPE OF ERROR.
11024		000064	MICPC=MICPC+1
11025	027510	061223	.WORD .S.
11026	027512		MOVE # 06, BREG ; LOAD FUNCTION CODE...
11027		000065	MICPC=MICPC+1
11028	027512	000406	.WORD .S.
11029	027514		MOVE BREG, OUT1 <CSR?> ; LOAD IT...
11030		000066	MICPC=MICPC+1
11031	027514	061227	.WORD .S.
11032	027516		CALL EROR ; ALU 2A W/C ERROR...
11033	027516		MOVE # <MICPC+3>, BREG
11034		000067	MICPC=MICPC+1
11035	027516	000471	.WORD .S.

11036	027520		SBR	EROR	
11037		000070	MICPC=MICPC+1		
11038	027520	104400	.WORD	.S.	
11039	027522		MOVE	SPAD <4>,MLR	;RESET DATA POINTER...
11040		000071	MICPC=MICPC+1		
11041	027522	070204	.WORD	.S.	
11042	027524		SBR	23	;LOOP ON ERROR...
11043		000072	MICPC=MICPC+1		
11044	027524	100453	.WORD	.S.	
11045	027526		3S: CALL	SCP1	
11046	027526		MOVE	# <MICPC+3>,BREG	
11047		000073	MICPC=MICPC+1		
11048	027526	000475	.WORD	.S.	
11049	027530		SBR	SCP1	
11050		000074	MICPC=MICPC+1		
11051	027530	104427	.WORD	.S.	
11052	027532		MOVE	SPAD <4>,MLR	;
11053		000075	MICPC=MICPC+1		
11054	027532	070204	.WORD	.S.	
11055	027534		SBR	23	;SCOPE THE DATA....
11056		000076	MICPC=MICPC+1		
11057	027534	100453	.WORD	.S.	
11058	027536		6S: MOVE	SPAD <4>,MLR	;RESET DATA POINTER...
11059		000077	MICPC=MICPC+1		
11060	027536	070204	.WORD	.S.	
11061	027540		MOVE	MEM,SPAD <0>,MARINC	;GET FIRST OPRAND...
11062		000100	MICPC=MICPC+1		
11063	027540	057220	.WORD	.S.	
11064	027542		MOVE	# 377,BREG	;
11065		000101	MICPC=MICPC+1		
11066	027542	000777	.WORD	.S.	
11067	027544		SADD	SPAD <2>,BREG	;SET C BIT...
11068		000102	MICPC=MICPC+1		
11069	027544	060402	.WORD	.S.!.DO	
11070	027546		SROL	SPAD <0>,BR.SP,MARINC	;SP 0 & BR = 2A W/C
11071		000103	MICPC=MICPC+1		
11072	027546	077540	.WORD	.S.	
11073	027550		MOVE	MEM,SPAD <3>,MARINC	;DUMMY INSTR, TO MARINC.
11074		000104	MICPC=MICPC+1		
11075	027550	057223	.WORD	.S.	
11076	027552		SIFEQ	MEM,SPAD <0> 9S	;BR IF GOOD...
11077					
11078					
11079	027552		SUB2C	SPAD <0>,MEM,NOP	
11080		000105	MICPC=MICPC+1		
11081	027552	040360	.WORD	.S.	
11082	027554		BZ	9S	
11083		000106	MICPC=MICPC+1		
11084	027554	101521	.WORD	.S.	
11085	027556		MOVE	MEM,OUT1 <CSR4>	;GOOD DATA.
11086		000107	MICPC=MICPC+1		
11087	027556	041224	.WORD	.S.	
11088	027560		MOVE	BREG,OUT1 <CSR5>	;BAD DATA.
11089		000110	MICPC=MICPC+1		
11090	027560	061225	.WORD	.S.	
11091	027562		MOVE	# 23,BREG	;ERROR TYPE...

11092		000111	MICPC=MICPC+1	
11093	027562	000423	.WORD .S.	
11094	027564		MOVE BREG,OUT1 <CSR3>	;
11095		000112	MICPC=MICPC+1	
11096	027564	061223	.WORD .S.	
11097	027566		MOVE # 06 BREG	;ALU FUNCTION CODE.
11098		000113	MICPC=MICPC+1	
11099	027566	000406	.WORD .S.	
11100	027570		MOVE BREG,OUT1 <CSR7>	;LOAD IT...
11101		000114	MICPC=MICPC+1	
11102	027570	061227	.WORD .S.	
11103	027572		CALL EROR ;REPORT ERROR...	
11104	027572		MOVE # <MICPC+3>,BREG	
11105		000115	MICPC=MICPC+1	
11106	027572	000517	.WORD .S.	
11107	027574		SBR EROR	
11108		000116	MICPC=MICPC+1	
11109	027574	104400	.WORD .S.	
11110	027576		MOVE SPAD <4>,MLR	;RESTORE DATA POINTER.
11111		000117	MICPC=MICPC+1	
11112	027576	070204	.WORD .S.	
11113	027600		SBR 6S	;LOOP ON ERROR...
11114		000120	MICPC=MICPC+1	
11115	027600	100477	.WORD .S.	
11116	027602		95: CALL SCPE	;SCOPE THE EROR...
11117	027602		MOVE # <MICPC+3>,BREG	
11118		000121	MICPC=MICPC+1	
11119	027602	000523	.WORD .S.	
11120	027604		SBR SCPE	
11121		000122	MICPC=MICPC+1	
11122	027604	104427	.WORD .S.	
11123	027606		MOVE SPAD <4>,MLR	;RESTORE DATA POINTER...
11124		000123	MICPC=MICPC+1	
11125	027606	070204	.WORD .S.	
11126	027610		SBR 6S	;SCOPE THE DATA...
11127		000124	MICPC=MICPC+1	
11128	027610	100477	.WORD .S.	
11129	027612		MOVE # 4,BREG	;UPDATE BACKGROUND POINTER.
11130		000125	MICPC=MICPC+1	
11131	027612	000404	.WORD .S.	
11132	027614		SADD BREG,SPAD <4>,MARINC	;ALSO DATA POINTER.
11133	027614	077004	.WORD .S. !MARINC!.DO	
11134	027616		SDEC SPAD <7>	;IS IT DONE??
11135		000126	MICPC=MICPC+1	
11136	027616	063167	.WORD .S.!.DSP	
11137	027620		BZ 4S	;YES, SCOPE THE TEST.
11138		000127	MICPC=MICPC+1	
11139	027620	101532	.WORD .S.	
11140	027622		SBR 2S	;DO, THE NEXT.
11141		000130	MICPC=MICPC+1	
11142	027622	100453	.WORD .S.	
11143	027624		45: CALL SCPE	;SCOPE THE TEST...
11144	027624		MOVE # <MICPC+3>,BREG	
11145		000131	MICPC=MICPC+1	
11146	027624	000533	.WORD .S.	
11147	027626		SBR SCPE	

```

11148          000132          MICPC=MICPC+1
11149 027626 104454          .WORD .S.
11150 027630          SBR 15          ;DO THE NEXT ITERATION...
11151          000133          MICPC=MICPC+1
11152 027630 100400          .WORD .S.
11153 027632          SALUT1 0,<SUB>,SUB1C,0,0,-1,-1,1,1,0,0,0,0,125,125,253,253,0,0,<A-B>,1,16
11154 027632          SXZ
11155
11156
11157
11158          ;***** TEST 52 *****
11159          ;*ALU TEST
11160          ;*TEST OF ALU FUNCTION SUB WITH C BIT CLEARED.
11161          ;*TEST OF ALU FUNCTION SUB WITH C BIT SET.
11162          ;*ALU FUNCTION (A-B)
11163          ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
11164 027632          SXZ          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
11165
11166          ;*****
11167 027632          STSTN
11168
11169          ; TEST 52
11170 027632 012737 000052 001202 TST52: MOV #52,STSTNM          ; LOAD THE NO. OF THIS TEST
11171 027640 012737 030160 001442 MOV #TST53,NEXT          ; POINT TO THE START OF NEXT TEST.
11172          ;R1 CONTAINS BASE KMC11 ADDRESS
11173 027646 004737 035536          JSR PC,LDVRWT          ;LOAD-VERIFY-WAIT.
11174 027652 027656          MCT52
11175 027654 104022          ERROR 22          ;TIME OUT ERROR...
11176 027656 012706 001200          MOV #STACK,SP          ;RESET STACK...
11177 027662 000177 151554          JMP @NEXT          ;GO TO NEXT TEST...
11178 027666          MCT52:
11179 027666          IS: MOVE #0,MLR          ;SET MAR+LO.
11180          MICPC=MICPC+1
11181 027666 010000          .WORD .S.
11182 027670          MOVE #0,MPR          ;SET MAR+HI.
11183          MICPC=MICPC+1
11184 027670 004000          .WORD .S.
11185 027672          MOVE #0,BREG          ;
11186          MICPC=MICPC+1
11187 027672 000400          .WORD .S.
11188 027674          MOVE BREG,SPAD <16>          ;FOR RETURN ADDRESS...
11189          MICPC=MICPC+1
11190 027674 063236          .WORD .S.
11191 027676          MOVE BREG,SPAD <0>          ;
11192          MICPC=MICPC+1
11193 027676 063220          .WORD .S.
11194 027700          MOVE BREG,SPAD <1>          ;
11195          MICPC=MICPC+1
11196 027700 063221          .WORD .S.
11197 027702          MOVE BREG,SPAD <2>          ;
11198          MICPC=MICPC+1
11199 027702 063222          .WORD .S.
11200 027704          SDEC SPAD <2>          ;
11201          MICPC=MICPC+1
11202 027704 063162          .WORD .S!.DSP
11203 027706          MOVE BREG,SPAD <4>          ;

```


11204		000010	MICPC=MICPC+1	
11205	027706	063224	.WORD .S.	
11206	027710		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
11207		000011	MICPC=MICPC+1	
11208	027710	016400	.WORD .S.	
11209	027712		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY
11210		000012	MICPC=MICPC+1	
11211	027712	016400	.WORD .S.	
11212	027714		MOVE # 0, MEM MARINC	
11213		000013	MICPC=MICPC+1	
11214	027714	016400	.WORD .S.	
11215	027716		MOVE # 0, MEM MARINC	;RESULT WITH C BIT SET.
11216		000014	MICPC=MICPC+1	
11217	027716	016400	.WORD .S.	
11218	027720		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
11219		000015	MICPC=MICPC+1	
11220	027720	016777	.WORD .S.	
11221	027722		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
11222		000016	MICPC=MICPC+1	
11223	027722	016400	.WORD .S.	
11224	027724		MOVE # -1, MEM MARINC	
11225		000017	MICPC=MICPC+1	
11226	027724	016777	.WORD .S.	
11227	027726		MOVE # -1, MEM MARINC	;RESULT WITH C BIT SET.
11228		000020	MICPC=MICPC+1	
11229	027726	016777	.WORD .S.	
11230	027730		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
11231		000021	MICPC=MICPC+1	
11232	027730	016400	.WORD .S.	
11233	027732		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
11234		000022	MICPC=MICPC+1	
11235	027732	016777	.WORD .S.	
11236	027734		MOVE # 1, MEM MARINC	
11237		000023	MICPC=MICPC+1	
11238	027734	016401	.WORD .S.	
11239	027736		MOVE # 1, MEM MARINC	;RESULT WITH C BIT SET.
11240		000024	MICPC=MICPC+1	
11241	027736	016401	.WORD .S.	
11242	027740		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY
11243		000025	MICPC=MICPC+1	
11244	027740	016777	.WORD .S.	
11245	027742		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
11246		000026	MICPC=MICPC+1	
11247	027742	016777	.WORD .S.	
11248	027744		MOVE # 0, MEM MARINC	
11249		000027	MICPC=MICPC+1	
11250	027744	016400	.WORD .S.	
11251	027746		MOVE # 0, MEM MARINC	;RESULT WITH C BIT SET.
11252		000030	MICPC=MICPC+1	
11253	027746	016400	.WORD .S.	
11254	027750		MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
11255		000031	MICPC=MICPC+1	
11256	027750	016525	.WORD .S.	
11257	027752		MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
11258		000032	MICPC=MICPC+1	
11259	027752	016525	.WORD .S.	

11260	027754		MOVE # 0, MEM MARINC	
11261		000033	MICPC=MICPC+1	
11262	027754	016400	.WORD .S.	
11263	027756		MOVE # 0, MEM MARINC ;RESULT WITH C BIT SET.	
11264		000034	MICPC=MICPC+1	
11265	027756	016400	.WORD .S.	
11266	027760		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
11267		000035	MICPC=MICPC+1	
11268	027760	016652	.WORD .S.	
11269	027762		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
11270		000036	MICPC=MICPC+1	
11271	027762	016525	.WORD .S.	
11272	027764		MOVE # 125, MEM MARINC	
11273		000037	MICPC=MICPC+1	
11274	027764	016525	.WORD .S.	
11275	027766		MOVE # 125, MEM MARINC ;RESULT WITH C BIT SET.	
11276		000040	MICPC=MICPC+1	
11277	027766	016525	.WORD .S.	
11278	027770		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
11279		000041	MICPC=MICPC+1	
11280	027770	016525	.WORD .S.	
11281	027772		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY	
11282		000042	MICPC=MICPC+1	
11283	027772	016652	.WORD .S.	
11284	027774		MOVE # 253, MEM MARINC	
11285		000043	MICPC=MICPC+1	
11286	027774	016653	.WORD .S.	
11287	027776		MOVE # 253, MEM MARINC ;RESULT WITH C BIT SET.	
11288		000044	MICPC=MICPC+1	
11289	027776	016653	.WORD .S.	
11290	030000		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
11291		000045	MICPC=MICPC+1	
11292	030000	016652	.WORD .S.	
11293	030002		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
11294		000046	MICPC=MICPC+1	
11295	030002	016652	.WORD .S.	
11296	030004		MOVE # 0, MEM MARINC ;LOAD THE DATA IN MEMORY.	
11297		000047	MICPC=MICPC+1	
11298	030004	016400	.WORD .S.	
11299	030006		MOVE # 0, MEM MARINC ;RESULT WITH C BIT SET.	
11300		000050	MICPC=MICPC+1	
11301	030006	016400	.WORD .S.	
11302	030010		MOVE # 7, SPAD <7> ;SET THE COUNT.	
11303		000051	MICPC=MICPC+1	
11304	030010	003007	.WORD .S.	
11305				
11306	030012		MOVE # 0, MLR ;MAR+0.	
11307		000052	MICPC=MICPC+1	
11308	030012	010000	.WORD .S.	
11309				
11310	030014		25: MOVE # 0, BREG ;	
11311		000053	MICPC=MICPC+1	
11312	030014	000400	.WORD .S.	
11313	030016		\$ADD SPAD <1>, BREG ;CLEAR C BIT.	
11314		000054	MICPC=MICPC+1	
11315	030016	060401	.WORD .S.!.DD	

11316	030020		MOVE MEM,SPAD <0> MARINC ;GET THE FIRST OPERAND.
11317		000055	MICPC=MICPC+1
11318	030020	057220	.WORD .S.
11319	030022		SUBIC MEM,SPAD <0>,BR.SP,MARINC ;SPAD <0>:=SFUNC.
11320		000056	MICPC=MICPC+1
11321	030022	057740	.WORD .S.
11322	030024		SIFEQ MEM,SPAD <0> 3S ;BRANCH IF GOOD.
11323			
11324			
11325	030024		SUB2C SPAD <0>,MEM,NOP
11326		000057	MICPC=MICPC+1
11327	030024	040360	.WORD .S.
11328	030026		BZ 3S
11329		000060	MICPC=MICPC+1
11330	030026	101473	.WORD .S.
11331	030030		MOVE MEM,OUT1 <4> ;GOOD DATA
11332		000061	MICPC=MICPC+1
11333	030030	041224	.WORD .S.
11334	030032		MOVE BREG,OUT1 <5> ;BAD DATA.
11335		000062	MICPC=MICPC+1
11336	030032	061225	.WORD .S.
11337	030034		MOVE # 15,BREG ;SET TYPE OF ERROR.
11338		000063	MICPC=MICPC+1
11339	030034	000415	.WORD .S.
11340	030036		MOVE BREG,OUT1 <3> ;SET TYPE OF ERROR.
11341		000064	MICPC=MICPC+1
11342	030036	061223	.WORD .S.
11343	030040		MOVE # 16,BREG ;LOAD FUNCTION CODE...
11344		000065	MICPC=MICPC+1
11345	030040	000416	.WORD .S.
11346	030042		MOVE BREG,OUT1 <CSR7> ;LOAD IT...
11347		000066	MICPC=MICPC+1
11348	030042	061227	.WORD .S.
11349	030044		CALL ERROR ;ALU SUB ERROR...
11350	030044		MOVE # <MICPC+3>,BREG
		000067	MICPC=MICPC+1
	030044	000471	.WORD .S.
	030046		SBR ERROR
			MICPC=MICPC+1
			.WORD .S.
			MOVE SPAD <4>,MLR ;RESET DATA POINTER...
			MICPC=MICPC+1
			.WORD .S.
			SBR 2S ;LOOP ON ERROR...
			MICPC=MICPC+1
		000072	.WORD .S.
		100453	
			3S:
			CALL SCP1
			MOVE # <MICPC+3>,BREG
			MICPC=MICPC+1
			.WORD .S.
			SBR SCP1
			MICPC=MICPC+1
			.WORD .S.
			MOVE SPAD <4>,MLR ;
			MICPC=MICPC+1
			.WORD .S.
11353	030054		
11354		000073	
11355	030054	000475	
11356	030056		
11357		000074	
11358	030056	104427	
11359	030060		
11370		000075	
11371	030060	070204	

11372	030062	000076	SBR 25 ; SCOPE THE DATA....
11373		100453	MICPC=MICPC+1
11374	030062		.WORD .S.
11375	030064		6S: MOVE SPAD <4>,MLR ;RESET DATA POINTER...
11376		000077	MICPC=MICPC+1
11377	030064	070204	.WORD .S.
11378	030066		MOVE MEM,SPAD <0>,MARINC ;GET FIRST OPRAND...
11379		000100	MICPC=MICPC+1
11380	030066	057220	.WORD .S.
11381	030070		MOVE # 377,BREG ;
11382		000101	MICPC=MICPC+1
11383	030070	000777	.WORD .S.
11384	030072		MOVE SPAD <2>,BREG ;SET C BIT...
11385		000102	MICPC=MICPC+1
11386	030072	060402	.WORD .S.!.DO
11387	030074		SUBIC MEM,SPAD <0>,BR.SP,MARINC ;SP 0 & BR = SUB
11388		000103	MICPC=MICPC+1
11389	030074	057740	.WORD .S.
11390	030076		MOVE MEM,SPAD <3>,MARINC ;DUMMY INSTR, TO MARINC.
11391		000104	MICPC=MICPC+1
11392	030076	057223	.WORD .S.
11393	030100		SIFEQ MEM,SPAD <0> 9S ;SR IF GOOD...
11394			
11395			
11396	030100		SUBC SPAD <0>,MEM,NOP
11397		000105	MICPC=MICPC+1
11398	030100	040360	.WORD .S.
11399	030102		BZ 9S
11400		000106	MICPC=MICPC+1
11401	030102	101521	.WORD .S.
11402	030104		MOVE MEM,OUT1 <CSR4> ;GOOD DATA.
11403		000107	MICPC=MICPC+1
11404	030104	041224	.WORD .S.
11405	030106		MOVE BREG,OUT1 <CSR5> ;BAD DATA.
11406		000110	MICPC=MICPC+1
11407	030106	061225	.WORD .S.
11408	030110		MOVE # 23,BREG ;ERROR TYPE...
11409		000111	MICPC=MICPC+1
11410	030110	000423	.WORD .S.
11411	030112		MOVE BREG,OUT1 <CSR3> ;
11412		000112	MICPC=MICPC+1
11413	030112	061223	.WORD .S.
11414	030114		MOVE # 16,BREG ;ALU FUNCTION CODE.
11415		000113	MICPC=MICPC+1
11416	030114	000416	.WORD .S.
11417	030116		MOVE BREG,OUT1 <CSR7> ;LOAD IT...
11418		000114	MICPC=MICPC+1
11419	030116	061227	.WORD .S.
11420	030120		CALL EROR ;REPORT ERROR...
11421	030120		MOVE # <MICPC+3>,BREG
11422		000115	MICPC=MICPC+1
11423	030120	000517	.WORD .S.
11424	030122		SBR EROR
11425		000116	MICPC=MICPC+1
11426	030122	104400	.WORD .S.
11427	030124		MOVE SPAD <4>,MLR ;RESTORE DATA POINTER.

```

11428      000117
11429      030124 070204
11430      030126
11431      000120
11432      030126 100477
11433      030130
11434      030130
95:        MICPC=MICPC+1
           .WORD .S.
           SBR 6S ;LOOP ON ERROR...
           MICPC=MICPC+1
           .WORD .S.
           CALL SCPE ;SCOPE THE ERROR...
           MOVE # <MICPC+3>,BREG
           MICPC=MICPC+1
           .WORD .S.
           SBR SCPE
           MICPC=MICPC+1
           .WORD .S.
           MOVE SPAD <4>,MLR ;RESTORE DATA POINTER...
           MICPC=MICPC+1
           .WORD .S.
           SBR 6S ;SCOPE THE DATA...
           MICPC=MICPC+1
           .WORD .S.
           MOVE # 4,BREG ;UPDATE BACKGROUND POINTER.
           MICPC=MICPC+1
           .WORD .S.
           SADD BREG,SPAD <4>,MARINC ;ALSO DATA POINTER.
           .WORD .S.!MARINC!.DO
           SDEC SPAD <7> ;IS IT DONE??
           MICPC=MICPC+1
           .WORD .S.!DSP
           BZ 45 ;YES, SCOPE THE TEST.
           MICPC=MICPC+1
           .WORD .S.
           SBR 2S ;DO, THE NEXT.
           MICPC=MICPC+1
           .WORD .S.
           CALL SCPE ;SCOPE THE TEST...
           MOVE # <MICPC+3>,BREG
           MICPC=MICPC+1
           .WORD .S.
           SBR SCPE
           MICPC=MICPC+1
           .WORD .S.
           SBR 1S ;DO THE NEXT ITERATION...
           MICPC=MICPC+1
           .WORD .S.
SALUT1    0,<ADD W/C>,ADDWC,0,1,-1,0,-1,0,376,-1,252,253,-1,0,-1,0,124,125,<A PLUS B PLUS
SXZ
           ***** TEST 53 *****
           *ALU TEST
           *TEST OF ALU FUNCTION ADD W/C WITH C BIT CLEARED.
           *TEST OF ALU FUNCTION ADD W/C WITH C BIT SET.
           *ALU FUNCTION (A PLUS B PLUS C)
           *LOAD MAIN MEMORY 16 WORDS OF DATA.
           *PERFORM THE FUNCTION, VERIFY THE RESULTS..
SXZ
           ;*****

```

```

11484 030160          STSTN
11485
11486          ; TEST 53
11487 030160 012737 000053 001202 TST53: MOV #53 STSTNM          ; LOAD THE NO. OF THIS TEST
11488 030166 012737 030506 001442      MOV #STST54,NEXT      ; POINT TO THE START OF NEXT TEST.
11489          ;R1 CONTAINS BASE KMC11 ADDRESS
11490 030174 004737 035536          JSR PC,LDVWNT        ;LOAD-VERIFY-WAIT.
11491 030200 030214          MCT53
11492 030202 104022          ERROR 22            ;TIME OUT ERROR...
11493 030204 012706 001200          MOV #STACK,SP        ;RESET STACK...
11494 030210 000177 151226          JMP @NEXT            ;GO TO NEXT TEST...
11495 030214
11496 030214
11497          MCT53:
11498          15: MOVE #0,MLR          ;SET MAR+LO.
11499          MICPC=MICPC+1
11500          .WORD .S.
11501          MOVE #0,MPR          ;SET MAR+HI.
11502          MICPC=MICPC+1
11503          .WORD .S.
11504          MOVE #0,BREG
11505          MICPC=MICPC+1          ;
11506          .WORD .S.
11507          MOVE #BREG,SPAD(16)    ;FOR RETURN ADDRESS...
11508          MICPC=MICPC+1
11509          .WORD .S.
11510          MOVE #BREG,SPAD(0)    ;
11511          MICPC=MICPC+1
11512          .WORD .S.
11513          MOVE #BREG,SPAD(1)    ;
11514          MICPC=MICPC+1
11515          .WORD .S.
11516          MOVE #BREG,SPAD(2)    ;
11517          MICPC=MICPC+1
11518          SDEC SPAD(2)          ;
11519          MICPC=MICPC+1
11520          .WORD .S.
11521          MOVE #BREG,SPAD(4)    ;
11522          MICPC=MICPC+1
11523          .WORD .S.
11524          MOVE #0,MEM MARINC    ;LOAD THE DATA IN MEMORY.
11525          MICPC=MICPC+1
11526          .WORD .S.
11527          MOVE #0,MEM MARINC    ;LOAD THE DATA IN MEMORY
11528          MICPC=MICPC+1
11529          .WORD .S.
11530          MOVE #0,MEM MARINC
11531          MICPC=MICPC+1
11532          .WORD .S.
11533          MOVE #1,MEM MARINC    ;RESULT WITH C BIT SET.
11534          MICPC=MICPC+1
11535          .WORD .S.
11536          MOVE #-1,MEM MARINC   ;LOAD THE DATA IN MEMORY.
11537          MICPC=MICPC+1
11538          .WORD .S.
11539          MOVE #0,MEM MARINC    ;LOAD THE DATA IN MEMORY.
          MICPC=MICPC+1

```

11540	030250	016400	.WORD .S.	
11541	030252		MOVE # -1, MEM MARINC	
11542		000017	MICPC=MICPC+1	
11543	030252	016777	.WORD .S.	
11544	030254		MOVE # 0, MEM MARINC ;RESULT WITH C BIT SET.	
11545		000020	MICPC=MICPC+1	
11546	030254	016400	.WORD .S.	
11547	030256		MOVE # 0, MEM MARINC ;LOAD THE DATA IN MEMORY.	
11548		000021	MICPC=MICPC+1	
11549	030256	016400	.WORD .S.	
11550	030260		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.	
11551		000022	MICPC=MICPC+1	
11552	030260	016777	.WORD .S.	
11553	030262		MOVE # -1, MEM MARINC	
11554		000023	MICPC=MICPC+1	
11555	030262	016777	.WORD .S.	
11556	030264		MOVE # 0, MEM MARINC ;RESULT WITH C BIT SET.	
11557		000024	MICPC=MICPC+1	
11558	030264	016400	.WORD .S.	
11559	030266		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY	
11560		000025	MICPC=MICPC+1	
11561	030266	016777	.WORD .S.	
11562	030270		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.	
11563		000026	MICPC=MICPC+1	
11564	030270	016777	.WORD .S.	
11565	030272		MOVE # 376, MEM MARINC	
11566		000027	MICPC=MICPC+1	
11567	030272	016776	.WORD .S.	
11568	030274		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.	
11569		000030	MICPC=MICPC+1	
11570	030274	016777	.WORD .S.	
11571	030276		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
11572		000031	MICPC=MICPC+1	
11573	030276	016525	.WORD .S.	
11574	030300		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
11575		000032	MICPC=MICPC+1	
11576	030300	016525	.WORD .S.	
11577	030302		MOVE # 252, MEM MARINC	
11578		000033	MICPC=MICPC+1	
11579	030302	016652	.WORD .S.	
11580	030304		MOVE # 253, MEM MARINC ;RESULT WITH C BIT SET.	
11581		000034	MICPC=MICPC+1	
11582	030304	016653	.WORD .S.	
11583	030306		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
11584		000035	MICPC=MICPC+1	
11585	030306	016652	.WORD .S.	
11586	030310		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
11587		000036	MICPC=MICPC+1	
11588	030310	016525	.WORD .S.	
11589	030312		MOVE # -1, MEM MARINC	
11590		000037	MICPC=MICPC+1	
11591	030312	016777	.WORD .S.	
11592	030314		MOVE # 0, MEM MARINC ;RESULT WITH C BIT SET.	
11593		000040	MICPC=MICPC+1	
11594	030314	016400	.WORD .S.	
11595	030316		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	

11596		06741		NICPC=NICPC+1	
11597	030316	016525		.WORD .S.	
11598	030320			MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY
11599		000042		NICPC=NICPC+1	
11600	030320	016652		.WORD .S.	
11601	030322			MOVE # -1, MEM MARINC	
11602		000043		NICPC=NICPC+1	
11603	030322	016777		.WORD .S.	
11604	030324			MOVE # 0, MEM MARINC	;RESULT WITH C BIT SET.
11605		000044		NICPC=NICPC+1	
11606	030324	016400		.WORD .S.	
11607	030326			MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY.
11608		000045		NICPC=NICPC+1	
11609	030326	016652		.WORD .S.	
11610	030330			MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY.
11611		000046		NICPC=NICPC+1	
11612	030330	016652		.WORD .S.	
11613	030332			MOVE # 124, MEM MARINC	;LOAD THE DATA IN MEMORY.
11614		000047		NICPC=NICPC+1	
11615	030332	016524		.WORD .S.	
11616	030334			MOVE # 125, MEM MARINC	;RESULT WITH C BIT SET.
11617		000050		NICPC=NICPC+1	
11618	030334	016525		.WORD .S.	
11619	030336			MOVE # 7, SPAD <7>	;SET THE COUNT.
11620		000051		NICPC=NICPC+1	
11621	030336	003007		.WORD .S.	
11622					
11623	030340			MOVE # 0, MLR	;MAR+0.
11624		000052		NICPC=NICPC+1	
11625	030340	010000		.WORD .S.	
11626					
11627	030342		25:	MOVE # 0, BREG	
11628		000053		NICPC=NICPC+1	
11629	030342	000400		.WORD .S.	
11630	030344			SADD SPAD <1>, BREG	;CLEAR C BIT.
11631		000054		NICPC=NICPC+1	
11632	030344	060401		.WORD .S. !.DD	
11633	030346			MOVE MEM, SPAD <0> MARINC	;GET THE FIRST OPERAND.
11634		000055		NICPC=NICPC+1	
11635	030346	057220		.WORD .S.	
11636	030350			ADDMC MEM, SPAD <0>, BR.SP, MARINC	;SPAD <0> := \$FUNC.
11637		000056		NICPC=NICPC+1	
11638	030350	057420		.WORD .S.	
11639	030352			SIFEQ MEM, SPAD <0> 3\$;BRANCH IF GOOD.
11640					
11641					
11642	030352			SUB2C SPAD <0>, MEM, NOP	
11643		000057		NICPC=NICPC+1	
11644	030352	040360		.WORD .S.	
11645	030354			BZ 3\$	
11646		000060		NICPC=NICPC+1	
11647	030354	101473		.WORD .S.	
11648	030356			MOVE MEM, OUT1 <4>	;GOOD DATA
11649		000061		NICPC=NICPC+1	
11650	030356	041224		.WORD .S.	
11651	030360			MOVE BREG, OUT1 <5>	;BAD DATA.

11652		000062	MICPC=MICPC+1	
11653	030360	061225	.WORD .S.	
11654	030362		MOVE # 15, BREG	;SET TYPE OF ERROR.
11655		000063	MICPC=MICPC+1	
11656	030362	000415	.WORD .S.	
11657	030364		MOVE BREG, OUT1 <3>	;SET TYPE OF ERROR.
11658		000064	MICPC=MICPC+1	
11659	030364	061223	.WORD .S.	
11660	030366		MOVE # 01, BREG	;LOAD FUNCTION CODE...
11661		000065	MICPC=MICPC+1	
11662	030366	000401	.WORD .S.	
11663	03037J		MOVE BREG, OUT1 <CSR7>	;LOAD IT...
11664		000066	MICPC=MICPC+1	
11665	030370	061227	.WORD .S.	
11666	030372		CALL EROR	;ALU ADD W/C ERROR...
11667	030372		MOVE # <MICPC+3>, BREG	
11668		000067	MICPC=MICPC+1	
11669	030372	000471	.WORD .S.	
11670	030374		SBR EROR	
11671		000070	MICPC=MICPC+1	
11672	030374	104400	.WORD .S.	
11673	030376		MOVE SPAD <4>, MLR	;RESET DATA POINTER...
11674		000071	MICPC=MICPC+1	
11675	030376	070204	.WORD .S.	
11676	030400		SBR 25	;LOOP ON ERROR...
11677		000072	MICPC=MICPC+1	
11678	030400	100453	.WORD .S.	
11679	030402		CALL SCP1	
11680	030402		MOVE # <MICPC+3>, BREG	
11681		000073	MICPC=MICPC+1	
11682	030402	000475	.WORD .S.	
11683	030404		SBR SCP1	
11684		000074	MICPC=MICPC+1	
11685	030404	104427	.WORD .S.	
11686	030406		MOVE SPAD <4>, MLR	;
11687		000075	MICPC=MICPC+1	
11688	030406	070204	.WORD .S.	
11689	030410		SBR 25	;SCOPE THE DATA....
11690		000076	MICPC=MICPC+1	
11691	030410	100453	.WORD .S.	
11692	030412		MOVE SPAD <4>, MLR	;RESET DATA POINTER...
11693		000077	MICPC=MICPC+1	
11694	030412	070204	.WORD .S.	
11695	030414		MOVE MEM, SPAD <0>, MARINC	;GET FIRST OPRAND...
11696		000100	MICPC=MICPC+1	
11697	030414	057220	.WORD .S.	
11698	030416		MOVE # 377, BREG	;
11699		000101	MICPC=MICPC+1	
11700	030416	000777	.WORD .S.	
11701	030420		SADD SPAD <2>, BREG	;SET C BIT...
11702		000102	MICPC=MICPC+1	
11703	030420	060402	.WORD .S. !. DO	
11704	030422		ADDMC MEM, SPAD <0>, BR.SP, MARINC	;SP 0 & BR = ADD W/C
11705		000103	MICPC=MICPC+1	
11706	030422	057420	.WORD .S.	
11707	030424		MOVE MEM, SPAD <3>, MARINC	;DUMMY INSTR, TO MARINC.

35:

65:

```

11708      000104      MICPC=MICPC+1
11709      030424      057223      .WORD      .S.
11710      030426      $IFEQ      MEM,SPAD <0>      9S      ;BR IF GOOD...
11711
11712
11713      030426      SUB2C      SPAD <0>,MEM,NOP
11714      000105      MICPC=MICPC+1
11715      030426      040360      .WORD      .S.
11716      030430      BZ      9S
11717      000106      MICPC=MICPC+1
11718      030430      101521      .WORD      .S.
11719      030432      MOVE      MEM,OUT1 <CSR4> ;GOOD DATA.
11720      000107      MICPC=MICPC+1
11721      030432      041224      .WORD      .S.
11722      030434      MOVE      BREG,OUT1 <CSR5>      ;BAD DATA.
11723      000110      MICPC=MICPC+1
11724      030434      061225      .WORD      .S.
11725      030436      MOVE      # 23,BREG      ;ERROR TYPE...
11726      000111      MICPC=MICPC+1
11727      030436      000423      .WORD      .S.
11728      030440      MOVE      BREG,OUT1 <CSR3>      ;
11729      000112      MICPC=MICPC+1
11730      030440      061223      .WORD      .S.
11731      030442      MOVE      # 01,BREG      ;ALU FUNCTION CODE.
11732      000113      MICPC=MICPC+1
11733      030442      000401      .WORD      .S.
11734      030444      MOVE      BREG,OUT1 <CSR7>      ;LOAD IT...
11735      000114      MICPC=MICPC+1
11736      030444      031227      .WORD      .S.
11737      030446      CALL      EROR ;REPORT ERROR...
11738      030446      MOVE      # <MICPC+3>,BREG
11739      000115      MICPC=MICPC+1
11740      030446      000517      .WORD      .S.
11741      030450      SBR      EROR
11742      000116      MICPC=MICPC+1
11743      030450      104400      .WORD      .S.
11744      030452      MOVE      SPAD <4>,MLR      ;RESTORE DATA POINTER.
11745      000117      MICPC=MICPC+1
11746      030452      070204      .WORD      .S.
11747      030454      SBR      6S      ;LOOP ON ERROR...
11748      000120      MICPC=MICPC+1
11749      030454      100477      .WORD      .S.
11750      030456      9S:      CALL      SCP1 ;SCOPE THE ERROR...
11751      030456      MOVE      # <MICPC+3>,BREG
11752      000121      MICPC=MICPC+1
11753      030456      000523      .WORD      .S.
11754      030460      SBR      SCP1
11755      000122      MICPC=MICPC+1
11756      030460      104427      .WORD      .S.
11757      030462      MOVE      SPAD <4>,MLR      ;RESTORE DATA POINTER...
11758      000123      MICPC=MICPC+1
11759      030462      070204      .WORD      .S.
11760      030464      SBR      6S      ;SCOPE THE DATA...
11761      000124      MICPC=MICPC+1
11762      030464      100477      .WORD      .S.
11763      030466      MOVE      # 4,BREG      ;UPDATE BACKGROUND POINTER.
    
```

```

11764      000125      MICPC=MICPC+1
11765      030466      000404      .WORD      $
11766      030470      $ADD      BREG,SPAD <4>,MARINC      ;ALSO DATA POINTER.
11767      030470      077004      .WORD      $,!MARINC!.D0
11768      030472      $DEC      SPAD <7>      ;IS IT DONE??
11769      000126      MICPC=MICPC+1
11770      030472      063167      .WORD      $,!DSP
11771      030474      BZ      45      ;YES, SCOPE THE TEST.
11772      000127      MICPC=MICPC+1
11773      030474      101532      .WORD      $
11774      030476      SBR      25      ;DO, THE NEXT.
11775      000130      MICPC=MICPC+1
11776      030476      100453      .WORD      $
11777      030500      45:      CALL      SCPE      ;SCOPE THE TEST...
11778      030500      MOVE      # <MICPC+3>,BREG
11779      000131      MICPC=MICPC+1
11780      030500      000533      .WORD      $
11781      030502      SBR      SCPE
11782      000132      MICPC=MICPC+1
11783      030502      104454      .WORD      $
11784      030504      SBR      15      ;DO THE NEXT ITERATION...
11785      000133      MICPC=MICPC+1
11786      030504      100400      .WORD      $
11787      030506      SALUT1  0,<SUB W/C>,SUBMC,-1,0,376,377,0,1,-1,0,-1,0,124,125,252,253,-1,0,<A-B-C>,1,02
11788      030506      SXZ
11789
11790
11791      ;***** TEST 54 *****
11792      ;*ALU TEST
11793      ;*TEST OF ALU FUNCTION SUB W/C WITH C BIT CLEARED.
11794      ;*TEST OF ALU FUNCTION SUB W/C WITH C BIT SET.
11795      ;*ALU FUNCTION (A-B-C)
11796      ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
11797      ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
11798      030506      SXZ
11799      ;:*****
11800
11801      030506      STSTN
11802      ; TEST 54
11803      ;-----
11804      030506      012737      000054      001202      TST54:  MOV      #54,STSTNM      ; LOAD THE NO. OF THIS TEST
11805      030514      012737      031034      001442      MOV      #TST55,NEXT      ; POINT TO THE START OF NEXT TEST.
11806      ;R1 CONTAINS BASE KMC11 ADDRESS
11807      030522      004737      035536      JSR      PC,LDRMT      ;LOAD-VERIFY-WAIT.
11808      030526      030542      MCT54
11809      030530      104022      ERROR      22      ;TIME OUT ERROR...
11810      030532      012706      001200      MOV      #STACK,SP      ;RESET STACK...
11811      030536      000177      150700      JMP      @NEXT      ;GO TO NEXT TEST...
11812      030542      MCT54:
11813      030542      1S:      MOVE      # 0,MLR      ;SET MAR+LO.
11814      000000      MICPC=MICPC+1
11815      030542      010000      .WORD      $
11816      030544      MOVE      # 0,MPR      ;SET MAR+HI.
11817      000001      MICPC=MICPC+1
11818      030544      004000      .WORD      $
11819      030546      MOVE      # 0,BREG

```

11820		000002	MICPC=MICPC+1	
11821	030546	000400	.WORD .S.	
11822	030550		MOVE BREG SPAD <16>	;FOR RETURN ADDRESS...
11823		000003	MICPC=MICPC+1	
11824	030550	063236	.WORD .S.	
11825	030552		MOVE BREG SPAD <0>	;
11826		000004	MICPC=MICPC+1	
11827	030552	063220	.WORD .S.	
11828	030554		MOVE BREG SPAD <1>	;
11829		000005	MICPC=MICPC+1	
11830	030554	063221	.WORD .S.	
11831	030556		MOVE BREG SPAD <2>	;
11832		000006	MICPC=MICPC+1	
11833	030556	063222	.WORD .S.	
11834	030560		SDEC SPAD <2>	;
11835		000007	MICPC=MICPC+1	
11836	030560	063162	.WORD .S.!.DSP	
11837	030562		MOVE BREG SPAD <4>	;
11838		000010	MICPC=MICPC+1	
11839	030562	063224	.WORD .S.	
11840	030564		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
11841		000011	MICPC=MICPC+1	
11842	030564	016400	.WORD .S.	
11843	030566		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY
11844		000012	MICPC=MICPC+1	
11845	030566	016400	.WORD .S.	
11846	030570		MOVE # -1, MEM MARINC	
11847		000013	MICPC=MICPC+1	
11848	030570	016777	.WORD .S.	
11849	030572		MOVE # 0, MEM MARINC	;RESULT WITH C BIT SET.
11850		000014	MICPC=MICPC+1	
11851	030572	016400	.WORD .S.	
11852	030574		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
11853		000015	MICPC=MICPC+1	
11854	030574	016777	.WORD .S.	
11855	030576		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
11856		000016	MICPC=MICPC+1	
11857	030576	016400	.WORD .S.	
11858	030600		MOVE # 376, MEM MARINC	
11859		000017	MICPC=MICPC+1	
11860	030600	016776	.WORD .S.	
11861	030602		MOVE # 377, MEM MARINC	;RESULT WITH C BIT SET.
11862		000020	MICPC=MICPC+1	
11863	030602	016777	.WORD .S.	
11864	030604		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
11865		000021	MICPC=MICPC+1	
11866	030604	016400	.WORD .S.	
11867	030606		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
11868		000022	MICPC=MICPC+1	
11869	030606	016777	.WORD .S.	
11870	030610		MOVE # 0, MEM MARINC	
11871		000023	MICPC=MICPC+1	
11872	030610	016400	.WORD .S.	
11873	030612		MOVE # 1, MEM MARINC	;RESULT WITH C BIT SET.
11874		000024	MICPC=MICPC+1	
11875	030612	016401	.WORD .S.	

11876	030614		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY
11877		000025	MICPC=MICPC+1
11878	030614	016777	.WORD .S.
11879	030616		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.
11880		000026	MICPC=MICPC+1
11881	030616	016777	.WORD .S.
11882	030620		MOVE # -1, MEM MARINC
11883		000027	MICPC=MICPC+1
11884	030620	016777	.WORD .S.
11885	030622		MOVE # 0, MEM MARINC ;RESULT WITH C BIT SET.
11886		000030	MICPC=MICPC+1
11887	030622	016400	.WORD .S.
11888	030624		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
11889		000031	MICPC=MICPC+1
11890	030624	016525	.WORD .S.
11891	030626		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
11892		000032	MICPC=MICPC+1
11893	030626	016525	.WORD .S.
11894	030630		MOVE # -1, MEM MARINC
11895		000033	MICPC=MICPC+1
11896	030630	016777	.WORD .S.
11897	030632		MOVE # 0, MEM MARINC ;RESULT WITH C BIT SET.
11898		000034	MICPC=MICPC+1
11899	030632	016400	.WORD .S.
11900	030634		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
11901		000035	MICPC=MICPC+1
11902	030634	016652	.WORD .S.
11903	030636		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
11904		000036	MICPC=MICPC+1
11905	030636	016525	.WORD .S.
11906	030640		MOVE # 124, MEM MARINC
11907		000037	MICPC=MICPC+1
11908	030640	016524	.WORD .S.
11909	030642		MOVE # 125, MEM MARINC ;RESULT WITH C BIT SET.
11910		000040	MICPC=MICPC+1
11911	030642	016525	.WORD .S.
11912	030644		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
11913		000041	MICPC=MICPC+1
11914	030644	016525	.WORD .S.
11915	030646		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY
11916		000042	MICPC=MICPC+1
11917	030646	016652	.WORD .S.
11918	030650		MOVE # 252, MEM MARINC
11919		000043	MICPC=MICPC+1
11920	030650	016652	.WORD .S.
11921	030652		MOVE # 253, MEM MARINC ;RESULT WITH C BIT SET.
11922		000044	MICPC=MICPC+1
11923	030652	016653	.WORD .S.
11924	030654		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
11925		000045	MICPC=MICPC+1
11926	030654	016652	.WORD .S.
11927	030656		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
11928		000046	MICPC=MICPC+1
11929	030656	016652	.WORD .S.
11930	030660		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.
11931		000047	MICPC=MICPC+1

11932	030660	016777	.WORD .S	
11933	030662		MOVE # 0 MEM MARINC	; RESULT WITH C BIT SET.
11934		000050	MICPC=MICPC+1	
11935	030662	016400	.WORD .S	
11936	030664		MOVE # 7 SPAD <7>	; SET THE COUNT.
11937		000051	MICPC=MICPC+1	
11938	030664	003007	.WORD .S	
11939				
11940	030666		MOVE # 0 MLR	; MAR+0.
11941		000052	MICPC=MICPC+1	
11942	030666	010000	.WORD .S	
11943				
11944	030670			
11945		000053	2S: MOVE # 0 BREG	
11946	030670	000400	MICPC=MICPC+1	
11947	030672		.WORD .S	
11948		000054	SADD SPAD <1>, BREG	; CLEAR C BIT.
11949	030672	060401	MICPC=MICPC+1	
11950	030674		.WORD .S ; DO	
11951		000055	MOVE MEM SPAD <0> MARINC	; GET THE FIRST OPERAND.
11952	030674	057220	MICPC=MICPC+1	
11953	030676		.WORD .S	
11954		000056	SUBC MEM SPAD <0>, BR.SP, MARINC	; SPAD <0> := \$FUNC.
11955	030676	057440	MICPC=MICPC+1	
11956	030700		.WORD .S	
11957			\$IFEQ MEM, SPAD <0> 3S	; BRANCH IF GOOD.
11958				
11959	030700			
11960		000057	SUBC SPAD <0>, MEM, NOP	
11961	030700	040360	MICPC=MICPC+1	
11962	030702		.WORD .S	
11963		000060	BZ 3S	
11964	030702	101473	MICPC=MICPC+1	
11965	030704		.WORD .S	
11966		000061	MOVE MEM, OUT1 <4>	; GOOD DATA
11967	030704	041224	MICPC=MICPC+1	
11968	030706		.WORD .S	
11969		000062	MOVE BREG, OUT1 <5>	; BAD DATA.
11970	030706	061225	MICPC=MICPC+1	
11971	030710		.WORD .S	
11972		000063	MOVE # 15 BREG	; SET TYPE OF ERROR.
11973	030710	000415	MICPC=MICPC+1	
11974	030712		.WORD .S	
11975		000064	MOVE BREG, OUT1 <3>	; SET TYPE OF ERROR.
11976	030712	061223	MICPC=MICPC+1	
11977	030714		.WORD .S	
11978		000065	MOVE # 02 BREG	; LOAD FUNCTION CODE...
11979	030714	000402	MICPC=MICPC+1	
11980	030716		.WORD .S	
11981		000066	MOVE BREG, OUT1 <CSR7>	; LOAD IT...
11982	030716	061227	MICPC=MICPC+1	
11983	030720		.WORD .S	
11984	030720		CALL EROR	; ALU SUB W/C ERROR...
11985		000067	MOVE # <MICPC+3>, BREG	
11986	030720	000471	MICPC=MICPC+1	
11987	030722		.WORD .S	
			SBR EROR	

11988		000070	MICPC=MICPC+1
11989	030722	104400	WORD S
11990	030724		MOVE SPAD <4>,MLR ;RESET DATA POINTER...
11991		000071	MICPC=MICPC+1
11992	030724	070204	WORD S
11993	030726		SBR 25 ;LOOP ON ERROR...
11994		000072	MICPC=MICPC+1
11995	030726	100453	WORD S
11996	030730		CALL SCP1
11997	030730		MOVE # <MICPC+3>,BREG
11998		000073	MICPC=MICPC+1
11999	030730	000475	WORD S
12000	030732		SBR SCP1
12001		000074	MICPC=MICPC+1
12002	030732	104427	WORD S
12003	030734		MOVE SPAD <4>,MLR ;
12004		000075	MICPC=MICPC+1
12005	030734	070204	WORD S
12006	030736		SBR 25 ;SCOPE THE DATA....
12007		000076	MICPC=MICPC+1
12008	030736	100453	WORD S
12009	030740		MOVE SPAD <4>,MLR ;RESET DATA POINTER...
12010		000077	MICPC=MICPC+1
12011	030740	070204	WORD S
12012	030742		MOVE MEM,SPAD <0>,MARINC ;GET FIRST OPRAND...
12013		000100	MICPC=MICPC+1
12014	030742	057220	WORD S
12015	030744		MOVE # 377,BREG ;
12016		000101	MICPC=MICPC+1
12017	030744	000777	WORD S
12018	030746		SBR SPAD <2>,BREG ;SET C BIT...
12019		000102	MICPC=MICPC+1
12020	030746	060402	WORD S ;.DD
12021	030750		SUBMC MEM,SPAD <0>,BR.SP,MARINC ;SP 0 & BR = SUB W/C
12022		000103	MICPC=MICPC+1
12023	030750	057440	WORD S
12024	030752		MOVE MEM,SPAD <3>,MARINC ;DUMMY INSTR, TO MARINC.
12025		000104	MICPC=MICPC+1
12026	030752	057223	WORD S
12027	030754		\$IFEQ MEM,SPAD <0> 95 ;BR IF GOOD...
12028			
12029			
12030	030754		SUB2C SPAD <0>,MEM,NOP
12031		000105	MICPC=MICPC+1
12032	030754	040360	WORD S
12033	030756		BZ 95
12034		000106	MICPC=MICPC+1
12035	030756	101521	WORD S
12036	030760		MOVE MEM,OUT1 <CSR4> ;GOOD DATA.
12037		000107	MICPC=MICPC+1
12038	030760	041224	WORD S
12039	030762		MOVE BREG,OUT1 <CSR5> ;BAD DATA.
12040		000110	MICPC=MICPC+1
12041	030762	061225	WORD S
12042	030764		MOVE # 23,BREG ;ERROR TYPE...
12043		000111	MICPC=MICPC+1

12044	030764	000423	.WORD .S.	
12045	030766		MOVE BREG OUT1 <CSR3>	;
12046		000112	MICPC=MICPC+1	
12047	030766	061223	.WORD .S.	
12048	030770		MOVE # 02, BREG	;ALU FUNCTION CODE.
12049		000113	MICPC=MICPC+1	
12050	030770	000402	.WORD .S.	
12051	030772		MOVE BREG OUT1 <CSR7>	;LOAD IT...
12052		000114	MICPC=MICPC+1	
12053	030772	061227	.WORD .S.	
12054	030774		CALL EROR	;REPORT ERROR...
12055	030774		MOVE # <MICPC+3>, BREG	
12056		000115	MICPC=MICPC+1	
12057	030774	000517	.WORD .S.	
12058	030776		SBR EROR	
12059		000116	MICPC=MICPC+1	
12060	030776	104400	.WORD .S.	
12061	031000		MOVE SPAD <4>, MLR	;RESTORE DATA POINTER.
12062		000117	MICPC=MICPC+1	
12063	031000	070204	.WORD .S.	
12064	031002		SBR 6S	;LOOP ON ERROR...
12065		000120	MICPC=MICPC+1	
12066	031002	100477	.WORD .S.	
12067	031004		CALL SCP1	;SCOPE THE ERROR...
12068	031004		MOVE # <MICPC+3>, BREG	
12069		000121	MICPC=MICPC+1	
12070	031004	000523	.WORD .S.	
12071	031006		SBR SCP1	
12072		000122	MICPC=MICPC+1	
12073	031006	104427	.WORD .S.	
12074	031010		MOVE SPAD <4>, MLR	;RESTORE DATA POINTER...
12075		000123	MICPC=MICPC+1	
12076	031010	070204	.WORD .S.	
12077	031012		SBR 6S	;SCOPE THE DATA...
12078		000124	MICPC=MICPC+1	
12079	031012	100477	.WORD .S.	
12080	031014		MOVE # 4, BREG	;UPDATE BACKGROUND POINTER.
12081		000125	MICPC=MICPC+1	
12082	031014	000404	.WORD .S.	
12083	031016		SADD BREG SPAD <4>, MARINC	;ALSO DATA POINTER.
12084	031016	077004	.WORD .S. !MARINC!.DO	
12085	031020		SDEC SPAD <7>	;IS IT DONE??
12086		000126	MICPC=MICPC+1	
12087	031020	063167	.WORD .S. !.DSP	
12088	031022		BZ 4S	;YES, SCOPE THE TEST.
12089		000127	MICPC=MICPC+1	
12090	031022	101532	.WORD .S.	
12091	031024		SBR 2S	;DO, THE NEXT.
12092		000130	MICPC=MICPC+1	
12093	031024	100453	.WORD .S.	
12094	031026		CALL SCPE	;SCOPE THE TEST...
12095	031026		MOVE # <MICPC+3>, BREG	
12096		000131	MICPC=MICPC+1	
12097	031026	000533	.WORD .S.	
12098	031030		SBR SCPE	
12099		000132	MICPC=MICPC+1	

12100	031030	104454				.WORD .S.	
12101	031032					SBR IS	;DO THE NEXT ITERATION...
12102		000133				MICPC=MICPC+1	
12103	031032	100400				WORD S	
12104	031034					SALUT1 0,<INC A>,\$INC,1,1,0,0,1,1,0,0,126,126,253,253,126,126,253,253,<A PLUS 1>,0,03	
12105	031034					SXZ	
12106							
12107							
12108						***** TEST 55 *****	
12109						*ALU TEST	
12110						*TEST OF ALU FUNCTION INC A WITH C BIT CLEARED.	
12111						*TEST OF ALU FUNCTION INC A WITH C BIT SET.	
12112						*ALU FUNCTION (A PLUS 1)	
12113						*LOAD MAIN MEMORY 16 WORDS OF DATA.	
12114						*PERFORM THE FUNCTION, VERIFY THE RESULTS..	
12115	031034					SXZ	
12116						*****	
12117							
12118	031034					STSTN	
12119							
12120						; TEST 55	
12121	031034	012737	000055	001202	TST55:	MOV #55,STSTNM	; LOAD THE NO. OF THIS TEST
12122	031042	012737	031362	001442		MOV #STST56,NEXT	; POINT TO THE START OF NEXT TEST.
12123							;RI CONTAINS BASE KMC11 ADDRESS
12124	031050	004737	035536			JSR PC,LDVWRT	;LOAD-VERIFY-WAIT.
12125	031054	031070				MCT55	
12126	031056	104022				ERROR 22	;TIME OUT ERROR...
12127	031060	012706	001200			MOV #STACK,SP	;RESET STACK.
12128	031064	000177	150352			JMP @NEXT	;GO TO NEXT TEST...
12129	031070				MCT55:		
12130	031070				IS:	MOVE #0,MLR	;SET MAR+LO.
12131		000000				MICPC=MICPC+1	
12132	031070	010000				.WORD .S.	
12133	031072					MOVE #0,MPR	;SET MAR+HI.
12134		000001				MICPC=MICPC+1	
12135	031072	004000				.WORD .S.	
12136	031074					MOVE #0,BREG	
12137		000002				MICPC=MICPC+1	
12138	031074	000400				.WORD .S.	
12139	031076					MOVE BREG,SPAD <16>	;FOR RETURN ADDRESS...
12140		000003				MICPC=MICPC+1	
12141	031076	063236				.WORD .S.	
12142	031100					MOVE BREG,SPAD <0>	;
12143		000004				MICPC=MICPC+1	
12144	031100	063220				.WORD .S.	
12145	031102					MOVE BREG,SPAD <1>	;
12146		000005				MICPC=MICPC+1	
12147	031102	063221				.WORD .S.	
12148	031104					MOVE BREG,SPAD <2>	;
12149		000006				MICPC=MICPC+1	
12150	031104	063222				.WORD .S.	
12151	031106					SDEC SPAD <2>	;
12152		000007				MICPC=MICPC+1	
12153	031106	063162				.S!.DSP	
12154	031110					MOVE BREG,SPAD <4>	;
12155		000010				MICPC=MICPC+1	

12156	031110	063224	.WORD .S.	
12157	031112		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
12158		000011	MICPC=MICPC+1	
12159	031112	016400	.WORD .S.	
12160	031114		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY
12161		000012	MICPC=MICPC+1	
12162	031114	016400	.WORD .S.	
12163	031116		MOVE # 1, MEM MARINC	
12164		000013	MICPC=MICPC+1	
12165	031116	016401	.WORD .S.	
12166	031120		MOVE # 1, MEM MARINC	;RESULT WITH C BIT SET.
12167		000014	MICPC=MICPC+1	
12168	031120	016401	.WORD .S.	
12169	031122		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
12170		000015	MICPC=MICPC+1	
12171	031122	016777	.WORD .S.	
12172	031124		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
12173		000016	MICPC=MICPC+1	
12174	031124	016400	.WORD .S.	
12175	031126		MOVE # 0, MEM MARINC	
12176		000017	MICPC=MICPC+1	
12177	031126	016400	.WORD .S.	
12178	031130		MOVE # 0, MEM MARINC	;RESULT WITH C BIT SET.
12179		000020	MICPC=MICPC+1	
12180	031130	016400	.WORD .S.	
12181	031132		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
12182		000021	MICPC=MICPC+1	
12183	031132	016400	.WORD .S.	
12184	031134		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
12185		000022	MICPC=MICPC+1	
12186	031134	016777	.WORD .S.	
12187	031136		MOVE # 1, MEM MARINC	
12188		000023	MICPC=MICPC+1	
12189	031136	016401	.WORD .S.	
12190	031140		MOVE # 1, MEM MARINC	;RESULT WITH C BIT SET.
12191		000024	MICPC=MICPC+1	
12192	031140	016401	.WORD .S.	
12193	031142		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY
12194		000025	MICPC=MICPC+1	
12195	031142	016777	.WORD .S.	
12196	031144		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
12197		000026	MICPC=MICPC+1	
12198	031144	016777	.WORD .S.	
12199	031146		MOVE # 0, MEM MARINC	
12200		000027	MICPC=MICPC+1	
12201	031146	016400	.WORD .S.	
12202	031150		MOVE # 0, MEM MARINC	;RESULT WITH C BIT SET.
12203		000030	MICPC=MICPC+1	
12204	031150	016400	.WORD .S.	
12205	031152		MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
12206		000031	MICPC=MICPC+1	
12207	031152	016525	.WORD .S.	
12208	031154		MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
12209		000032	MICPC=MICPC+1	
12210	031154	016525	.WORD .S.	
12211	031156		MOVE # 126, MEM MARINC	

12212		000033	MICPC=MICPC+1	
12213	031156	016526	.WORD .S.	
12214	031160		MOVE # 126, MEM MARINC	;RESULT WITH C BIT SET.
12215		000034	MICPC=MICPC+1	
12216	031160	016526	.WORD .S.	
12217	031162		MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY.
12218		000035	MICPC=MICPC+1	
12219	031162	016652	.WORD .S.	
12220	031164		MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
12221		000036	MICPC=MICPC+1	
12222	031164	016525	.WORD .S.	
12223	031166		MOVE # 253, MEM MARINC	
12224		000037	MICPC=MICPC+1	
12225	031166	016653	.WORD .S.	
12226	031170		MOVE # 253, MEM MARINC	;RESULT WITH C BIT SET.
12227		000040	MICPC=MICPC+1	
12228	031170	016653	.WORD .S.	
12229	031172		MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
12230		000041	MICPC=MICPC+1	
12231	031172	016525	.WORD .S.	
12232	031174		MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY
12233		000042	MICPC=MICPC+1	
12234	031174	016652	.WORD .S.	
12235	031176		MOVE # 126, MEM MARINC	
12236		000043	MICPC=MICPC+1	
12237	031176	016526	.WORD .S.	
12238	031200		MOVE # 126, MEM MARINC	;RESULT WITH C BIT SET.
12239		000044	MICPC=MICPC+1	
12240	031200	016526	.WORD .S.	
12241	031202		MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY.
12242		000045	MICPC=MICPC+1	
12243	031202	016652	.WORD .S.	
12244	031204		MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY.
12245		000046	MICPC=MICPC+1	
12246	031204	016652	.WORD .S.	
12247	031206		MOVE # 253, MEM MARINC	;LOAD THE DATA IN MEMORY.
12248		000047	MICPC=MICPC+1	
12249	031206	016653	.WORD .S.	
12250	031210		MOVE # 253, MEM MARINC	;RESULT WITH C BIT SET.
12251		000050	MICPC=MICPC+1	
12252	031210	016653	.WORD .S.	
12253	031212		MOVE # 7, SPAD <7>	;SET THE COUNT.
12254		000051	MICPC=MICPC+1	
12255	031212	003007	.WORD .S.	
12256				
12257	031214		MOVE # 0, MLR	;MAR+0.
12258		000052	MICPC=MICPC+1	
12259	031214	010000	.WORD .S.	
12260				
12261	031216		25: MOVE # 0, BREG	;
12262		000053	MICPC=MICPC+1	
12263	031216	000400	.WORD .S.	
12264	031220		SADD SPAD <1>, BREG	;CLEAR C BIT.
12265		000054	MICPC=MICPC+1	
12266	031220	060401	.WORD .S. ! DO	
12267	031222		MOVE MEM, SPAD <0> MARINC	;GET THE FIRST OPERAND.

12268		000055	MICPC=MICPC+1
12269	031222	057220	.WORD .S.
12270	031224		\$INC SPAD <0>,BR.SP, MARINC ;
12271		000056	MICPC=MICPC+1
12272	031224	077460	.WORD .S.
12273	031226		\$IFEQ MEM,SPAD <0> 3\$;BRANCH IF GOOD.
12274			
12275			
12276	031226		SUB2C SPAD <0>,MEM,NOP
12277		000057	MICPC=MICPC+1
12278	031226	040360	.WORD .S.
12279	031230		BZ 3\$
12280		000060	MICPC=MICPC+1
12281	031230	101473	.WORD .S.
12282	031232		MOVE MEM,OUT1 <4> ;GOOD DATA
12283		000061	MICPC=MICPC+1
12284	031232	041224	.WORD .S.
12285	031234		MOVE BREG,OUT1 <5> ;BAD DATA.
12286		000062	MICPC=MICPC+1
12287	031234	061225	.WORD .S.
12288	031236		MOVE #15,BREG ;SET TYPE OF ERROR.
12289		000063	MICPC=MICPC+1
12290	031236	000415	.WORD .S.
12291	031240		MOVE BREG,OUT1 <3> ;SET TYPE OF ERROR.
12292		000064	MICPC=MICPC+1
12293	031240	061223	.WORD .S.
12294	031242		MOVE #03,BREG ;LOAD FUNCTION CODE...
12295		000065	MICPC=MICPC+1
12296	031242	000403	.WORD .S.
12297	031244		MOVE BREG,OUT1 <CSR7> ;LOAD IT...
12298		000066	MICPC=MICPC+1
12299	031244	061227	.WORD .S.
12300	031246		CALL ENOR ;ALU INC: A ERROR...
12301	031246		MOVE # <MICPC+3>,BREG
12302		000067	MICPC=MICPC+1
12303	031246	000471	.WORD .S.
12304	031250		SBR ENOR
12305		000070	MICPC=MICPC+1
12306	031250	104400	.WORD .S.
12307	031252		MOVE SPAD <4>,MLR ;RESET DATA POINTER...
12308		000071	MICPC=MICPC+1
12309	031252	070204	.WORD .S.
12310	031254		SBR 2\$;LOOP ON ERROR...
12311		000072	MICPC=MICPC+1
12312	031254	100453	.WORD .S.
12313	031256		CALL SCP1
12314	031256		MOVE # <MICPC+3>,BREG
12315		000073	MICPC=MICPC+1
12316	031256	000475	.WORD .S.
12317	031260		SBR SCP1
12318		000074	MICPC=MICPC+1
12319	031260	104427	.WORD .S.
12320	031262		MOVE SPAD <4>,MLR ;
12321		000075	MICPC=MICPC+1
12322	031262	070204	.WORD .S.
12323	031264		SBR 2\$;SCOPE THE DATA....

3\$:

12324		000076	MICPC=MICPC+1
12325	031264	100453	.WORD .S.
12326	031266		6S: MOVE SPAD <4>,MLR ;RESET DATA POINTER...
12327		000077	MICPC=MICPC+1
12328	031266	070204	.WORD .S.
12329	031270		MOVE MEM,SPAD <0>,MARINC ;GET FIRST OPRAND...
12330		000100	MICPC=MICPC+1
12331	031270	057220	.WORD .S.
12332	031272		MOVE # 377,BREG ;
12333		000101	MICPC=MICPC+1
12334	031272	000777	.WORD .S.
12335	031274		SADD SPAD <2>,BREG ;SET C BIT...
12336		000102	MICPC=MICPC+1
12337	031274	060402	.WORD .S. .00
12338	031276		SINC SPAD <0>,BR.SP,MARINC ;SP 0 & BR = INC A
12339		000103	MICPC=MICPC+1
12340	031276	077460	.WORD .S.
12341	031300		MOVE MEM,SPAD <3>,MARINC ;DUMMY INSTR, TO MARINC.
12342		000104	MICPC=MICPC+1
12343	031300	057223	.WORD .S.
12344	031302		SIFEQ MEM,SPAD <0> 9S ;BR IF GOOD...
12345			
12346			
12347	031302		SUB2C SPAD <0>,MEM,NOP
12348		000105	MICPC=MICPC+1
12349	031302	040360	.WORD .S.
12350	031304		BZ 9S
12351		000106	MICPC=MICPC+1
12352	031304	101521	.WORD .S.
12353	031306		MOVE MEM,OUT1 <CSR4> ;GOOD DATA.
12354		000107	MICPC=MICPC+1
12355	031306	041224	.WORD .S.
12356	031310		MOVE BREG,OUT1 <CSR5> ;BAD DATA.
12357		000110	MICPC=MICPC+1
12358	031310	061225	.WORD .S.
12359	031312		MOVE # 23,BREG ;ERROR TYPE...
12360		000111	MICPC=MICPC+1
12361	031312	000423	.WORD .S.
12362	031314		MOVE BREG,OUT1 <CSR3> ;
12363		000112	MICPC=MICPC+1
12364	031314	061223	.WORD .S.
12365	031316		MOVE # 03,BREG ;ALU FUNCTION CODE.
12366		000113	MICPC=MICPC+1
12367	031316	000403	.WORD .S.
12368	031320		MOVE BREG,OUT1 <CSR7> ;LOAD IT...
12369		000114	MICPC=MICPC+1
12370	031320	061227	.WORD .S.
12371	031322		CALL EROR ;REPORT ERROR...
12372	031322		MOVE # <MICPC+3>,BREG
12373		000115	MICPC=MICPC+1
12374	031322	000517	.WORD .S.
12375	031324		SBR EROR
12376		000116	MICPC=MICPC+1
12377	031324	104400	.WORD .S.
12378	031326		MOVE SPAD <4>,MLR ;RESTORE DATA POINTER.
12379		000117	MICPC=MICPC+1

12380 031326 070204
 12381 031330
 12382 000120
 12383 031330 100477
 12384 031332
 12385 031332
 12386 000121
 12387 031332 000523
 12388 031334
 12389 000122
 12390 031334 104427
 12391 031336
 12392 000123
 12393 031336 070204
 12394 031340
 12395 000124
 12396 031340 100477
 12397 031342
 12398 000125
 12399 031342 000404
 12400 031344
 12401 031344 077004
 12402 031346
 12403 000126
 12404 031346 063167
 12405 031350
 12406 000127
 12407 031350 101532
 12408 031352
 12409 000130
 12410 031352 100453
 12411 031354
 12412 031354
 12413 000131
 12414 031354 000533
 12415 031356
 12416 000132
 12417 031356 104454
 12418 031360
 12419 000133
 12420 031360 100400
 12421 031362
 12422 031362
 12423
 12424
 12425
 12426
 12427
 12428
 12429
 12430
 12431
 12432 031362
 12433
 12434
 12435 031362

```

      .WORD      .S.
      SBR        6S
      MICPC=MICPC+1
      ;LOOP ON ERROR...
95:   .WORD      .S.
      CALL       SCP1
      MOVE       # <MICPC+3>,BREG ;SCOPE THE ERROR...
      MICPC=MICPC+1
      .WORD      .S.
      SBR        SCP1
      MICPC=MICPC+1
      .WORD      .S.
      MOVE       SPAD <4>,MLR    ;RESTORE DATA POINTER...
      MICPC=MICPC+1
      .WORD      .S.
      SBR        6S
      MICPC=MICPC+1
      ;SCOPE THE DATA...
      .WORD      .S.
      MOVE       # 4,BREG
      MICPC=MICPC+1
      ;UPDATE BACKGROUND POINTER.
      .WORD      .S.
      SADD       BREG,SPAD <4>,MARINC ;ALSO DATA POINTER.
      .WORD      .S. !MARINC!.D0
      SDEC       SPAD <7>
      MICPC=MICPC+1
      ;IS IT DONE??
      .WORD      .S. !.DSP
      BZ         4S
      MICPC=MICPC+1
      ;YES, SCOPE THE TEST.
      .WORD      .S.
      SBR        2S
      MICPC=MICPC+1
      ;DO, THE NEXT.
45:   .WORD      .S.
      CALL       SCPE
      MOVE       # <MICPC+3>,BREG ;SCOPE THE TEST...
      MICPC=MICPC+1
      .WORD      .S.
      SBR        SCPE
      MICPC=MICPC+1
      .WORD      .S.
      SBR        1S
      MICPC=MICPC+1
      ;DO THE NEXT ITERATION...
      .WORD      .S.
      $ALUT1    0,<2A>,$ASL,0,0,376,376,0,0,376,376,252,252,124,124,252,252,124,124,<A PLUS A>,0
      $XZ
      ;***** TEST 56 *****
      ;*ALU TEST
      ;*TEST OF ALU FUNCTION 2A WITH C BIT CLEARED.
      ;*TEST OF ALU FUNCTION 2A WITH C BIT SET.
      ;*ALU FUNCTION (A PLUS A)
      ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
      ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
      ;*****
      $XZ
      $TSTN
  
```

```

12436 ; TEST 56
12437 -----
12438 031362 012737 000056 001202 TST56: MOV #56,STSTNM ; LOAD THE NO. OF THIS TEST
12439 031370 012737 031710 001442 MOV #TST57,NEXT ; POINT TO THE START OF NEXT TEST.
12440 ;R1 CONTAINS BASE KMC11 ADDRESS
12441 031376 004737 035536 JSR PC,LDRWT ;LOAD-VERIFY-WAIT.
12442 031402 031416 MCT56
12443 031404 104022 ERROR 22 ;TIME OUT ERROR...
12444 031406 012706 001200 MOV #STACK,SP ;RESET STACK...
12445 031412 000177 150024 JMP @NEXT ;GO TO NEXT TEST...
12446 031416 MCT56:
12447 15: MOVE #0,MLR ;SET MAR+LO.
12448 000000 MICPC=MICPC+1
12449 031416 010000 .WORD .S.
12450 031420 MOVE #0,MPR ;SET MAR+HI.
12451 000001 MICPC=MICPC+1
12452 031420 004000 .WORD .S.
12453 031422 MOVE #0,BREG ;
12454 000002 MICPC=MICPC+1
12455 031422 000400 .WORD .S.
12456 031424 MOVE BREG,SPAD (16) ;FOR RETURN ADDRESS...
12457 000003 MICPC=MICPC+1
12458 031424 063236 .WORD .S.
12459 031426 MOVE BREG,SPAD (0) ;
12460 000004 MICPC=MICPC+1
12461 031426 063220 .WORD .S.
12462 031430 MOVE BREG,SPAD (1) ;
12463 000005 MICPC=MICPC+1
12464 031430 063221 .WORD .S.
12465 031432 MOVE BREG,SPAD (2) ;
12466 000006 MICPC=MICPC+1
12467 031432 063222 .WORD .S.
12468 031434 SDEC SPAD (2) ;
12469 000007 MICPC=MICPC+1
12470 031434 063162 .WORD .S...DSP
12471 031436 MOVE BREG,SPAD (4) ;
12472 000010 MICPC=MICPC+1
12473 031436 063224 .WORD .S.
12474 031440 MOVE #0,MEM MARINC ;LOAD THE DATA IN MEMORY.
12475 000011 MICPC=MICPC+1
12476 031440 016400 .WORD .S.
12477 031442 MOVE #0,MEM MARINC ;LOAD THE DATA IN MEMORY
12478 000012 MICPC=MICPC+1
12479 031442 016400 .WORD .S.
12480 031444 MOVE #0,MEM MARINC
12481 000013 MICPC=MICPC+1
12482 031444 016400 .WORD .S.
12483 031446 MOVE #0,MEM MARINC ;RESULT WITH C BIT SET.
12484 000014 MICPC=MICPC+1
12485 031446 016400 .WORD .S.
12486 031450 MOVE #-1,MEM MARINC ;LOAD THE DATA IN MEMORY.
12487 000015 MICPC=MICPC+1
12488 031450 016777 .WORD .S.
12489 031452 MOVE #0,MEM MARINC ;LOAD THE DATA IN MEMORY.
12490 000016 MICPC=MICPC+1
12491 031452 016400 .WORD .S.

```

12492	031454	000017	MOVE # 376, MEM MARINC	
12493		016776	MICPC=MICPC+1	
12494	031454	016776	.WORD .S.	
12495	031456	000020	MOVE # 376, MEM MARINC	;RESULT WITH C BIT SET.
12496		016776	MICPC=MICPC+1	
12497	031456	000021	.WORD .S.	
12498	031460	016400	MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
12499		016400	MICPC=MICPC+1	
12500	031460	016400	.WORD .S.	
12501	031462	000022	MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
12502		016777	MICPC=MICPC+1	
12503	031462	000023	.WORD .S.	
12504	031464	016400	MOVE # 0, MEM MARINC	
12505		016400	MICPC=MICPC+1	
12506	031466	000024	.WORD .S.	
12507		016400	MOVE # 0, MEM MARINC	;RESULT WITH C BIT SET.
12508		016400	MICPC=MICPC+1	
12509	031466	000024	.WORD .S.	
12510	031470	016400	MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY
12511		000025	MICPC=MICPC+1	
12512	031470	016777	.WORD .S.	
12513	031472	000026	MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
12514		016777	MICPC=MICPC+1	
12515	031472	016777	.WORD .S.	
12516	031474	000027	MOVE # 376, MEM MARINC	
12517		016776	MICPC=MICPC+1	
12518	031474	016776	.WORD .S.	
12519	031476	000030	MOVE # 376, MEM MARINC	;RESULT WITH C BIT SET.
12520		016776	MICPC=MICPC+1	
12521	031476	016776	.WORD .S.	
12522	031500	000031	MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
12523		016525	MICPC=MICPC+1	
12524	031500	016525	.WORD .S.	
12525	031502	000032	MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
12526		016525	MICPC=MICPC+1	
12527	031502	016525	.WORD .S.	
12528	031504	000033	MOVE # 252, MEM MARINC	
12529		016652	MICPC=MICPC+1	
12530	031504	016652	.WORD .S.	
12531	031506	000034	MOVE # 252, MEM MARINC	;RESULT WITH C BIT SET.
12532		016652	MICPC=MICPC+1	
12533	031506	016652	.WORD .S.	
12534	031510	000035	MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY.
12535		016652	MICPC=MICPC+1	
12536	031510	016652	.WORD .S.	
12537	031512	000036	MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
12538		016525	MICPC=MICPC+1	
12539	031512	016525	.WORD .S.	
12540	031514	000037	MOVE # 124, MEM MARINC	
12541		016524	MICPC=MICPC+1	
12542	031514	016524	.WORD .S.	
12543	031516	000040	MOVE # 124, MEM MARINC	;RESULT WITH C BIT SET.
12544		016524	MICPC=MICPC+1	
12545	031516	016524	.WORD .S.	
12546	031520	000041	MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
12547		000041	MICPC=MICPC+1	

12548	031520	016525	.WORD .S.	
12549	031522		MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY
12550		000042	MICPC=MICPC+1	
12551	031522	016652	.WORD .S.	
12552	031524		MOVE # 252, MEM MARINC	
12553		000043	MICPC=MICPC+1	
12554	031524	016652	.WORD .S.	
12555	031526		MOVE # 252, MEM MARINC	;RESULT WITH C BIT SET.
12556		000044	MICPC=MICPC+1	
12557	031526	016652	.WORD .S.	
12558	031530		MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY.
12559		000045	MICPC=MICPC+1	
12560	031530	016652	.WORD .S.	
12561	031532		MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY.
12562		000046	MICPC=MICPC+1	
12563	031532	016652	.WORD .S.	
12564	031534		MOVE # 124, MEM MARINC	;LOAD THE DATA IN MEMORY.
12565		000047	MICPC=MICPC+1	
12566	031534	016524	.WORD .S.	
12567	031536		MOVE # 124, MEM MARINC	;RESULT WITH C BIT SET.
12568		000050	MICPC=MICPC+1	
12569	031536	016524	.WORD .S.	
12570	031540		MOVE # 7 SPAD (7)	;SET THE COUNT.
12571		000051	MICPC=MICPC+1	
12572	031540	003007	.WORD .S.	
12573				
12574	031542		MOVE # 0, MLR	;MAR+0.
12575		000052	MICPC=MICPC+1	
12576	031542	010000	.WORD .S.	
12577				
12578	031544		25: MOVE # 0, BREG	
12579		000053	MICPC=MICPC+1	
12580	031544	000400	.WORD .S.	
12581	031546		SAD SPAD (1), BREG	;CLEAR C BIT.
12582		000054	MICPC=MICPC+1	
12583	031546	060401	.WORD .S. !.DD	
12584	031550		MOVE MEM, SPAD (0) MARINC	;GET THE FIRST OPERAND.
12585		000055	MICPC=MICPC+1	
12586	031550	057220	.WORD .S.	
12587	031552		SASL SPAD (0), BR. SP, MARINC	
12588		000056	MICPC=MICPC+1	
12589	031552	077520	.WORD .S.	
12590	031554		SIFEQ MEM, SPAD (0) 3S	;BRANCH IF GOOD.
12591				
12592				
12593	031554		SUB2C SPAD (0), MEM, NOP	
12594		000057	MICPC=MICPC+1	
12595	031554	040360	.WORD .S.	
12596	031556		BZ 3S	
12597		000060	MICPC=MICPC+1	
12598	031556	101473	.WORD .S.	
12599	031560		MOVE MEM, OUT1 (4)	;GOOD DATA
12600		000061	MICPC=MICPC+1	
12601	031560	041224	.WORD .S.	
12602	031562		MOVE BREG, OUT1 (5)	;BAD DATA.
12603		000062	MICPC=MICPC+1	

12604	031562	061225	.WORD .S.	
12605	031564		MOVE # 15, BREG	;SET TYPE OF ERROR.
12606		000063	MICPC=MICPC+1	
12607	031564	000415	.WORD .S.	
12608	031566		MOVE BREG, OUT1 <3>	;SET TYPE OF ERROR.
12609		000064	MICPC=MICPC+1	
12610	031566	061223	.WORD .S.	
12611	031570		MOVE # 05, BREG	;LOAD FUNCTION CODE...
12612		000065	MICPC=MICPC+1	
12613	031570	000405	.WORD .S.	
12614	031572		MOVE BREG, OUT1 <CSR7>	;LOAD IT...
12615		000066	MICPC=MICPC+1	
12616	031572	061227	.WORD .S.	
12617	031574		CALL EROR	;ALU 2A ERROR...
12618	031574		MOVE # <MICPC+3>, BREG	
12619		000067	MICPC=MICPC+1	
12620	031574	000471	.WORD .S.	
12621	031576		SBR EROR	
12622		000070	MICPC=MICPC+1	
12623	031576	104400	.WORD .S.	
12624	031600		MOVE SPAD <4>, MLR	;RESET DATA POINTER...
12625		000071	MICPC=MICPC+1	
12626	031600	070204	.WORD .S.	
12627	031602		SBR 25	;LOOP ON ERROR...
12628		000072	MICPC=MICPC+1	
12629	031602	100453	.WORD .S.	
12630	031604		CALL SCP1	
12631	031604		MOVE # <MICPC+3>, BREG	
12632		000073	MICPC=MICPC+1	
12633	031604	000475	.WORD .S.	
12634	031606		SBR SCP1	
12635		000074	MICPC=MICPC+1	
12636	031606	104427	.WORD .S.	
12637	031610		MOVE SPAD <4>, MLR	;
12638		000075	MICPC=MICPC+1	
12639	031610	070204	.WORD .S.	
12640	031612		SBR 25	;SCOPE THE DATA....
12641		000076	MICPC=MICPC+1	
12642	031612	100453	.WORD .S.	
12643	031614		MOVE SPAD <4>, MLR	;RESET DATA POINTER...
12644		000077	MICPC=MICPC+1	
12645	031614	070204	.WORD .S.	
12646	031616		MOVE MEM, SPAD <0>, MARINC	;GET FIRST OPRAND...
12647		000100	MICPC=MICPC+1	
12648	031616	057220	.WORD .S.	
12649	031620		MOVE # 377, BREG	;
12650		000101	MICPC=MICPC+1	
12651	031620	000777	.WORD .S.	
12652	031622		SADD SPAD <2>, BREG	;SET C BIT...
12653		000102	MICPC=MICPC+1	
12654	031622	060402	.WORD .S. !.DU	
12655	031624		SASL SPAD <0>, BR.SP, MARINC	;SP 0 & BR = 2A
12656		000103	MICPC=MICPC+1	
12657	031624	077520	.WORD .S.	
12658	031626		MOVE MEM, SPAD <3>, MARINC	;DUMMY INSTR, TO MARINC.
12659		000104	MICPC=MICPC+1	

```

12660 031626 057223      .WORD      .S.
12661 031630      $IFEQ     MEM,SPAD <0>    9$      ;BR IF GOOD...
12662
12663
12664 031630      SUB2C     SPAD <0>,MEM,NOP
12665      MICPC=MICPC+1
12666 031630 040360      .WORD      .S.
12667 031632      BZ       9$
12668      MICPC=MICPC+1
12669 031632 101521      .WORD      .S.
12670 031634      MOVE     MEM,OUT1 <CSR4> ;GOOD DATA.
12671      MICPC=MICPC+1
12672 031634 041224      .WORD      .S.
12673 031636      MOVE     BREG,OUT1 <CSR5> ;BAD DATA.
12674      MICPC=MICPC+1
12675 031636 061225      .WORD      .S.
12676 031640      MOVE     # 23,BREG      ;ERROR TYPE...
12677      MICPC=MICPC+1
12678 031640 000423      .WORD      .S.
12679 031642      MOVE     BREG,OUT1 <CSR3> ;
12680      MICPC=MICPC+1
12681 031642 061223      .WORD      .S.
12682 031644      MOVE     # 05,BREG      ;ALU FUNCTION CODE.
12683      MICPC=MICPC+1
12684 031644 000405      .WORD      .S.
12685 031646      MOVE     BREG,OUT1 <CSR7> ;LOAD IT...
12686      MICPC=MICPC+1
12687 031646 061227      .WORD      .S.
12688 031650      CALL     EROR ;REPORT ERROR...
12689 031650      MOVE     # <MICPC+3>,BREG
12690      MICPC=MICPC+1
12691 031650 000517      .WORD      .S.
12692 031652      SBR     EROR
12693      MICPC=MICPC+1
12694 031652 104400      .WORD      .S.
12695 031654      MOVE     SPAD <4>,MLR      ;RESTORE DATA POINTER.
12696      MICPC=MICPC+1
12697 031654 070204      .WORD      .S.
12698 031656      SBR     6$      ;LOOP ON ERROR...
12699      MICPC=MICPC+1
12700 031656 100477      .WORD      .S.
12701 031660      CALL     SCP1 ;SCOPE THE ERROR...
12702 031660      MOVE     # <MICPC+3>,BREG
12703      MICPC=MICPC+1
12704 031660 000523      .WORD      .S.
12705 031662      SBR     SCP1
12706      MICPC=MICPC+1
12707 031662 104427      .WORD      .S.
12708 031664      MOVE     SPAD <4>,MLR      ;RESTORE DATA POINTER...
12709      MICPC=MICPC+1
12710 031664 070204      .WORD      .S.
12711 031666      SBR     6$      ;SCOPE THE DATA...
12712      MICPC=MICPC+1
12713 031666 100477      .WORD      .S.
12714 031670      MOVE     # 4,BREG      ;UPDATE BACKGROUND POINTER.
12715      MICPC=MICPC+1

```

12716	031670	000404			WORD	S			
12717	031672				SADD	BREG, SPAD (4), MARINC			; ALSO DATA POINTER.
12718	031672	077004			WORD	S	!MARINC!.DO		
12719	031674				SDEC	SPAD (7)			; IS IT DONE??
12720		000126			MICPC=MICPC+1				
12721	031674	063167			.WORD	S	!.DSP		
12722	031676				BZ	45			; YES, SCOPE THE TEST.
12723		000127			MICPC=MICPC+1				
12724	031676	101532			.WORD	S			
12725	031700				SBR	25			; DO, THE NEXT.
12726		000130			MICPC=MICPC+1				
12727	031700	100453			.WORD	S			
12728	031702			45:	CALL	SCPE			; SCOPE THE TEST...
12729	031702				MOVE	# <MICPC+3>, BREG			
12730		000131			MICPC=MICPC+1				
12731	031702	000533			.WORD	S			
12732	031704				SBR	SCPE			
12733		000132			MICPC=MICPC+1				
12734	031704	104454			.WORD	S			
12735	031706				SBR	15			; DO THE NEXT ITERATION...
12736		000133			MICPC=MICPC+1				
12737	031706	100403			.WORD	S			
12738	031710				SALUT1	0, <A PLUS C>, SADC, 0, 1, -1, 0, 0, 1, -1, 0, 125, 126, 252, 253, 125, 126, 252, 253, <A PLUS C>, 0			
12739	031710				SXZ				
12740									
12741									
12742									
12743									
12744									
12745									
12746									
12747									
12748									
12749	031710				SXZ				
12750									
12751									
12752	031710				STSTN				
12753									
12754									
12755	031710	012737	000057	001202	TST57:	MOV	#57, STSTN		; LOAD THE N THIS TEST
12756	031716	012737	032236	001442		MOV	#ST56, NEXT		; POINT TO 1 RT OF NEXT TEST.
12757									; R1 CONTAINS BASE KN... ADDRESS
12758	031724	004737	035536			JSR	PC, LDVANT		; LOAD-VERIFY-WAIT.
12759	031730	031744				MCT57			
12760	031732	104022				ERROR	22		; TIME OUT ERROR...
12761	031734	012706	001200			MOV	#STACK, SP		; RESET STACK...
12762	031740	000177	147476			JMP	#NEXT		; GO TO NEXT TEST...
12763	031744				MCT57:				
12764	031744				IS:	MOVE	# 0, MLR		; SET MAR+LO.
12765		000000				MICPC=MICPC+1			
12766	031744	010000				.WORD	S		
12767	031746					MOVE	# 0, MPR		; SET MAR+HI.
12768		000001				MICPC=MICPC+1			
12769	031746	004000				.WORD	S		
12770	031750					MOVE	# 0, BREG		
12771		000002				MICPC=MICPC+1			

```

***** TEST 57 *****
*ALU TEST
*TEST OF ALU FUNCTION A PLUS C WITH C BIT CLEARED.
*TEST OF ALU FUNCTION A PLUS C WITH C BIT SET.
*ALU FUNCTION (A PLUS C)
*LOAD MAIN MEMORY 16 WORDS OF DATA.
*PERFORM THE FUNCTION, VERIFY THE RESULTS..

```

```

*****
; TEST 57
-----
MOV #57, STSTN ; LOAD THE N THIS TEST
MOV #ST56, NEXT ; POINT TO 1 RT OF NEXT TEST.
; R1 CONTAINS BASE KN... ADDRESS
JSR PC, LDVANT ; LOAD-VERIFY-WAIT.
MCT57
ERROR 22 ; TIME OUT ERROR...
MOV #STACK, SP ; RESET STACK...
JMP #NEXT ; GO TO NEXT TEST...

```

12772	031750	000400	.WORD .S.	
12773	031752		MOVE #REG SPAD <16>	;FOR RETURN ADDRESS...
12774		000003	MICPC=MICPC+1	
12775	031752	063236	.WORD .S.	
12776	031754		MOVE #REG SPAD <0>	;
12777		000004	MICPC=MICPC+1	
12778	031754	063220	.WORD .S.	
12779	031756		MOVE #REG SPAD <1>	;
12780		000005	MICPC=MICPC+1	
12781	031756	063221	.WORD .S.	
12782	031760		MOVE #REG SPAD <2>	;
12783		000006	MICPC=MICPC+1	
12784	031760	063222	.WORD .S.	
12785	031762		SDC SPAD <2>	;
12786		000007	MICPC=MICPC+1	
12787	031762	063162	.WORD .S.!.DSP	
12788	031764		MOVE #REG SPAD <4>	;
12789		000010	MICPC=MICPC+1	
12790	031764	063224	.WORD .S.	
12791	031766		MOVE # 0 MEM MARINC	;LOAD THE DATA IN MEMORY.
12792		000011	MICPC=MICPC+1	
12793	031766	016400	.WORD .S.	
12794	031770		MOVE # 0 MEM MARINC	;LOAD THE DATA IN MEMORY
12795		000012	MICPC=MICPC+1	
12796	031770	016400	.WORD .S.	
12797	031772		MOVE # 0 MEM MARINC	
12798		000013	MICPC=MICPC+1	
12799	031772	016400	.WORD .S.	
12800	031774		MOVE # 1 MEM MARINC	;RESULT WITH C BIT SET.
12801		000014	MICPC=MICPC+1	
12802	031774	016401	.WORD .S.	
12803	031776		MOVE # -1 MEM MARINC	;LOAD THE DATA IN MEMORY.
12804		000015	MICPC=MICPC+1	
12805	031776	016777	.WORD .S.	
12806	032000		MOVE # 0 MEM MARINC	;LOAD THE DATA IN MEMORY.
12807		000016	MICPC=MICPC+1	
12808	032000	016400	.WORD .S.	
12809	032002		MOVE # -1 MEM MARINC	
12810		000017	MICPC=MICPC+1	
12811	032002	016777	.WORD .S.	
12812	032004		MOVE # 0 MEM MARINC	;RESULT WITH C BIT SET.
12813		000020	MICPC=MICPC+1	
12814	032004	016400	.WORD .S.	
12815	032006		MOVE # 0 MEM MARINC	;LOAD THE DATA IN MEMORY.
12816		000021	MICPC=MICPC+1	
12817	032006	016400	.WORD .S.	
12818	032010		MOVE # -1 MEM MARINC	;LOAD THE DATA IN MEMORY.
12819		000022	MICPC=MICPC+1	
12820	032010	016777	.WORD .S.	
12821	032012		MOVE # 0 MEM MARINC	
12822		000023	MICPC=MICPC+1	
12823	032012	016400	.WORD .S.	
12824	032014		MOVE # 1 MEM MARINC	;RESULT WITH C BIT SET.
12825		000024	MICPC=MICPC+1	
12826	032014	016401	.WORD .S.	
12827	032016		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY

12828		000025	MICPC=MICPC+1	
12829	032016	016777	.WORD .S.	
12830	032020		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.	
12831		000026	MICPC=MICPC+1	
12832	032020	016777	.WORD .S.	
12833	032022		MOVE # -1, MEM MARINC	
12834		000027	MICPC=MICPC+1	
12835	032022	016777	.WORD .S.	
12836	032024		MOVE # 0, MEM MARINC ;RESULT WITH C 9IT SET.	
12837		000030	MICPC=MICPC+1	
12838	032024	016400	.WORD .S.	
12839	032026		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
12840		000031	MICPC=MICPC+1	
12841	032026	016525	.WORD .S.	
12842	032030		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
12843		000032	MICPC=MICPC+1	
12844	032030	016525	.WORD .S.	
12845	032032		MOVE # 125, MEM MARINC	
12846		000033	MICPC=MICPC+1	
12847	032032	016525	.WORD .S.	
12848	032034		MOVE # 126, MEM MARINC ;RESULT WITH C BIT SET.	
12849		000034	MICPC=MICPC+1	
12850	032034	016526	.WORD .S.	
12851	032036		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
12852		000035	MICPC=MICPC+1	
12853	032036	016652	.WORD .S.	
12854	032040		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
12855		000036	MICPC=MICPC+1	
12856	032040	016525	.WORD .S.	
12857	032042		MOVE # 252, MEM MARINC	
12858		000037	MICPC=MICPC+1	
12859	032042	016652	.WORD .S.	
12860	032044		MOVE # 253, MEM MARINC ;RESULT WITH C BIT SET.	
12861		000040	MICPC=MICPC+1	
12862	032044	016653	.WORD .S.	
12863	032046		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
12864		000041	MICPC=MICPC+1	
12865	032046	016525	.WORD .S.	
12866	032050		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY	
12867		000042	MICPC=MICPC+1	
12868	032050	016652	.WORD .S.	
12869	032052		MOVE # 125, MEM MARINC	
12870		000043	MICPC=MICPC+1	
12871	032052	016525	.WORD .S.	
12872	032054		MOVE # 126, MEM MARINC ;RESULT WITH C BIT SET.	
12873		000044	MICPC=MICPC+1	
12874	032054	016526	.WORD .S.	
12875	032056		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
12876		000045	MICPC=MICPC+1	
12877	032056	016652	.WORD .S.	
12878	032060		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
12879		000046	MICPC=MICPC+1	
12880	032060	016652	.WORD .S.	
12881	032062		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
12882		000047	MICPC=MICPC+1	
12883	032062	016652	.WORD .S.	

12884	032064		MOVE # 253, MEM MARINC ; RESULT WITH C BIT SET.
12885		000050	MICPC=MICPC+1
12886	032064	016653	.WORD .S.
12887	032066		MOVE # 7, SPAD <7> ; SET THE COUNT.
12888		000051	MICPC=MICPC+1
12889	032066	003007	.WORD .S.
12890			
12891	032070		MOVE # 0, MLR ; MAR+0.
12892		000052	MICPC=MICPC+1
12893	032070	010000	.WORD .S.
12894			
12895	032072		25: MOVE # 0, BREG ;
12896		000053	MICPC=MICPC+1
12897	032072	000400	.WORD .S.
12898	032074		SADD SPAD <1>, BREG ; CLEAR C BIT.
12899		000054	MICPC=MICPC+1
12900	032074	060401	.WORD .S. !. DO
12901	032076		MOVE MEM, SPAD <0> MARINC ; GET THE FIRST OPERAND.
12902		000055	MICPC=MICPC+1
12903	032076	057220	.WORD .S.
12904	032100		SADC SPAD <0>, BR. SP, MARINC ;
12905		000056	MICPC=MICPC+1
12906	032100	077500	.WORD .S.
12907	032102		SIFEQ MEM, SPAD <0> 3S ; BRANCH IF GOOD.
12908			
12909			
12910	032102		SU32C SPAD <0>, MEM, NOP
12911		000057	MICPC=MICPC+1
12912	032102	040360	.WORD .S.
12913	032104		BZ 3S
12914		000060	MICPC=MICPC+1
12915	032104	101473	.WORD .S.
12916	032106		MOVE MEM, OUT1 <4> ; GOOD DATA
12917		000061	MICPC=MICPC+1
12918	032106	041224	.WORD .S.
12919	032110		MOVE BREG, OUT1 <5> ; BAD DATA.
12920		000062	MICPC=MICPC+1
12921	032110	061225	.WORD .S.
12922	032112		MOVE # 15, BREG ; SET TYPE OF ERROR.
12923		000063	MICPC=MICPC+1
12924	032112	000415	.WORD .S.
12925	032114		MOVE BREG, OUT1 <3> ; SET TYPE OF ERROR.
12926		000064	MICPC=MICPC+1
12927	032114	061223	.WORD .S.
12928	032116		MOVE # 04, BREG ; LOAD FUNCTION CODE...
12929		000065	MICPC=MICPC+1
12930	032116	000404	.WORD .S.
12931	032120		MOVE BREG, OUT1 <CSR7> ; LOAD IT...
12932		000066	MICPC=MICPC+1
12933	032120	061227	.WORD .S.
12934	032122		CALL EROR ; ALU A PLUS C ERROR...
12935	032122		MOVE # <MICPC+3>, BREG
12936		000067	MICPC=MICPC+1
12937	032122	000471	.WORD .S.
12938	032124		SBR EROR
12939		000070	MICPC=MICPC+1

12942 032124 104400
 12943 032126 000071
 12944 032126 070204
 12945 032130 000072
 12946 032130 100453
 12947 032132
 12948 032132
 12949 000073
 12950 032132 000475
 12951 032134
 12952 000074
 12953 032134 104427
 12954 032136
 12955 000075
 12956 032136 070204
 12957 032140
 12958 000076
 12959 032140 100453
 12960 032142
 12961 000077
 12962 032142 070204
 12963 032144
 12964 000100
 12965 032144 057220
 12966 032146
 12967 000101
 12968 032146 000777
 12969 032150
 12970 000102
 12971 032150 060402
 12972 032152
 12973 000103
 12974 032152 077500
 12975 032154
 12976 000104
 12977 032154 057223
 12978 032156
 12979
 12980
 12981 032156
 12982 000105
 12983 032156 040360
 12984 032160
 12985 000106
 12986 032160 101521
 12987 032162
 12988 000107
 12989 032162 041224
 12990 032164
 12991 000110
 12992 032164 061225
 12993 032166
 12994 000111
 12995 032166 000423

```

    .WORD .S.
    MOVE SPAD <4>,MLR ;RESET DATA POINTER...
    MICPC=MICPC+1
    .WORD .S.
    SBR 23 ;LOOP ON ERROR...
    MICPC=MICPC+1
    .WORD .S.
    3S: CALL SCP1
    MOVE # <MICPC+3>,BREG
    MICPC=MICPC+1
    .WORD .S.
    SBR SCP1
    MICPC=MICPC+1
    .WORD .S.
    MOVE SPAD <4>,MLR ;
    MICPC=MICPC+1
    .WORD .S.
    SBR 23 ;SCOPE THE DATA....
    MICPC=MICPC+1
    .WORD .S.
    6S: MOVE SPAD <4>,MLR ;RESET DATA POINTER...
    MICPC=MICPC+1
    .WORD .S.
    MOVE MEM,SPAD <0>,MARINC ;GET FIRST OPRAND...
    MICPC=MICPC+1
    .WORD .S.
    MOVE # 377,BREG ;
    MICPC=MICPC+1
    .WORD .S.
    SADD SPAD <2>,BREG ;SET C BIT...
    MICPC=MICPC+1
    .WORD .S.
    SADC SPAD <0>,BR.SP,MARINC ;SP 0 & BR = A PLUS C
    MICPC=MICPC+1
    .WORD .S.
    MOVE MEM,SPAD <3>,MARINC ;DUMMY INSTR, TO MARINC.
    MICPC=MICPC+1
    .WORD .S.
    SIFEQ 9S ;BR IF GOOD...

    SUB2C SPAD <0>,MEM,NOP
    MICPC=MICPC+1
    .WORD .S.
    BZ 9S
    MICPC=MICPC+1
    .WORD .S.
    MOVE MEM,OUT1 <CSR4> ;GOOD DATA.
    MICPC=MICPC+1
    .WORD .S.
    MOVE BREG,OUT1 <CSR5> ;BAD DATA.
    MICPC=MICPC+1
    .WORD .S.
    MOVE # 23,BREG ;ERROR TYPE...
    MICPC=MICPC+1
    .WORD .S.
  
```


12996	032170		MOVE BREG OUT1 <CSR3>	;
12997		000112	NICPC=NICPC+1	
12998	032170	061223	.WORD .S.	
12999	032172		MOVE # 04 BREG	;ALU FUNCTION CODE.
13000		000113	NICPC=NICPC+1	
13001	032172	000404	.WORD .S.	
13002	032174		MOVE BREG OUT1 <CSR7>	;LOAD IT...
13003		000114	NICPC=NICPC+1	
13004	032174	061227	.WORD .S.	
13005	032176		CALL ENOR REPORT ERROR...	
13006	032176		MOVE # <NICPC+3>,BREG	
13007		000115	NICPC=NICPC+1	
13008	032176	000517	.WORD .S.	
13009	032200		SBR ENOR	
13010		000116	NICPC=NICPC+1	
13011	032200	104400	.WORD .S.	
13012	032202		MOVE SPAD <4>,MLR	;RESTORE DATA POINTER.
13013		000117	NICPC=NICPC+1	
13014	032202	070204	.WORD .S.	
13015	032204		SBR #5	;LOOP ON ERROR...
13016		000120	NICPC=NICPC+1	
13017	032204	100477	.WORD .S.	
13018	032206		CALL SCP1	;SCOPE THE ERROR...
13019	032206		MOVE # <NICPC+3>,BREG	
13020		000121	NICPC=NICPC+1	
13021	032206	000523	.WORD .S.	
13022	032210		SBR SCP1	
13023		000122	NICPC=NICPC+1	
13024	032210	104427	.WORD .S.	
13025	032212		MOVE SPAD <4>,MLR	;RESTORE DATA POINTER...
13026		000123	NICPC=NICPC+1	
13027	032212	070204	.WORD .S.	
13028	032214		SBR #5	;SCOPE THE DATA...
13029		000124	NICPC=NICPC+1	
13030	032214	100477	.WORD .S.	
13031	032216		MOVE # 4 BREG	;UPDATE BACKGROUND POINTER.
13032		000125	NICPC=NICPC+1	
13033	032216	000404	.WORD .S.	
13034	032220		SADD BREG SPAD <4> MARINC	;ALSO DATA POINTER.
13035	032220	077004	.WORD .S. !MARINC!.DD	
13036	032222		SDEC SPAD <7>	;IS IT DONE??
13037		000126	NICPC=NICPC+1	
13038	032222	063167	.WORD .S.!.DSP	
13039	032224		BZ 45	;YES, SCOPE THE TEST.
13040		000127	NICPC=NICPC+1	
13041	032224	101532	.WORD .S.	
13042	032226		SBR 25	;DO, THE NEXT.
13043		000130	NICPC=NICPC+1	
13044	032226	100453	.WORD .S.	
13045	032230		CALL SCPE	;SCOPE THE TEST...
13046	032230		MOVE # <NICPC+3>,BREG	
13047		000131	NICPC=NICPC+1	
13048	032230	000533	.WORD .S.	
13049	032232		SBR SCPE	
13050		000132	NICPC=NICPC+1	
13051	032232	104454	.WORD .S.	

```

13052 032234          SBR 15          ;DO THE NEXT ITERATION...
13053          000133      MICPC=MICPC+1
13054 032234          .WORD 100400
13055 032236          SALUT1 0,(2'S COMP SUB),SUB2C,-1,-1,376,376,0,0,-1,-1,-1,-1,124,124,252,252,-1,-1,(A-B-
13056 032236          SXZ
13057
13058
13059          ;***** TEST 60 *****
13060          ;ALU TEST
13061          ;TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT CLEARED.
13062          ;TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT SET.
13063          ;ALU FUNCTION (A-B-1)
13064          ;LOAD MAIN MEMORY 16 WORDS OF DATA.
13065          ;PERFORM THE FUNCTION, VERIFY THE RESULTS..
13066 032236          SXZ
13067          ;*****
13068
13069 032236          STSTN
13070          ; TEST 60
13071          ;-----
13072 032236 012737 000060 001202 TST60: MOV #60,STSTN          ; LOAD THE NO. OF THIS TEST
13073 032244 012737 032564 001442      MOV #TST61,NEXT      ; POINT TO THE START OF NEXT TEST.
13074          ;R1 CONTAINS BASE KMC11 ADDRESS
13075 032252 004737 035536          JSR PC,LDRINT      ;LOAD-VERIFY-WAIT.
13076 032256 032272          MCT60
13077 032260 104022          ERROR 22          ;TIME OUT ERROR...
13078 032262 012706 001200      MOV #STACK,SP      ;RESET STACK...
13079 032266 000177 147150      JMP @NEXT          ;GO TO NEXT TEST...
13080 032272
13081 032272          MCT60:
13082          15: MOVE #0,MLR          ;SET MAR+LO.
13083 032272 010000      MICPC=MICPC+1
13084 032274          .WORD .S.
13085          MOVE #0,MPR          ;SET MAR+HI.
13086 032274 004000      MICPC=MICPC+1
13087 032276          .WORD .S.
13088          MOVE #0,BREG
13089 032276 000400      MICPC=MICPC+1
13090 032300          .WORD .S.
13091          MOVE BREG,SPAD <16>      ;FOR RETURN ADDRESS...
13092 032300 063236      MICPC=MICPC+1
13093 032302          .WORD .S.
13094          MOVE BREG,SPAD <0>      ;
13095 032302 063220      MICPC=MICPC+1
13096 032304          .WORD .S.
13097          MOVE BREG,SPAD <1>      ;
13098 032304 063221      MICPC=MICPC+1
13099 032306          .WORD .S.
13100          MOVE BREG,SPAD <2>      ;
13101 032306 063222      MICPC=MICPC+1
13102 032310          .WORD .S.
13103          SDEC SPAD <2>          ;
13104 032310 063162      MICPC=MICPC+1
13105 032312          .WORD .S.
13106          MOVE BREG,SPAD <4>      ;
13107 032312 063224      MICPC=MICPC+1
13108          .WORD .S.

```

13108	032314		MOVE # 0, MEM MARINC ;LOAD THE DATA IN MEMORY.
13109		000011	MICPC=MICPC+1
13110	032314	016400	.WORD .S.
13111	032316		MOVE # 0, MEM MARINC ;LOAD THE DATA IN MEMORY
13112		000012	MICPC=MICPC+1
13113	032316	016400	.WORD .S.
13114	032320		MOVE # -1, MEM MARINC
13115		000013	MICPC=MICPC+1
13116	032320	016777	.WORD .S.
13117	032322		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.
13118		000014	MICPC=MICPC+1
13119	032322	016777	.WORD .S.
13120	032324		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.
13121		000015	MICPC=MICPC+1
13122	032324	016777	.WORD .S.
13123	032326		MOVE # 0, MEM MARINC ;LOAD THE DATA IN MEMORY.
13124		000016	MICPC=MICPC+1
13125	032326	016400	.WORD .S.
13126	032330		MOVE # 376, MEM MARINC
13127		000017	MICPC=MICPC+1
13128	032330	016776	.WORD .S.
13129	032332		MOVE # 376, MEM MARINC ;RESULT WITH C BIT SET.
13130		000020	MICPC=MICPC+1
13131	032332	016776	.WORD .S.
13132	032334		MOVE # 0, MEM MARINC ;LOAD THE DATA IN MEMORY.
13133		000021	MICPC=MICPC+1
13134	032334	016400	.WORD .S.
13135	032336		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.
13136		000022	MICPC=MICPC+1
13137	032336	016777	.WORD .S.
13138	032340		MOVE # 0, MEM MARINC
13139		000023	MICPC=MICPC+1
13140	032340	016400	.WORD .S.
13141	032342		MOVE # 0, MEM MARINC ;RESULT WITH C BIT SET.
13142		000024	MICPC=MICPC+1
13143	032342	016400	.WORD .S.
13144	032344		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY
13145		000025	MICPC=MICPC+1
13146	032344	016777	.WORD .S.
13147	032346		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.
13148		000026	MICPC=MICPC+1
13149	032346	016777	.WORD .S.
13150	032350		MOVE # -1, MEM MARINC
13151		000027	MICPC=MICPC+1
13152	032350	016777	.WORD .S.
13153	032352		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.
13154		000030	MICPC=MICPC+1
13155	032352	016777	.WORD .S.
13156	032354		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
13157		000031	MICPC=MICPC+1
13158	032354	016525	.WORD .S.
13159	032356		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
13160		000032	MICPC=MICPC+1
13161	032356	016525	.WORD .S.
13162	032360		MOVE # -1, MEM MARINC
13163		000033	MICPC=MICPC+1

13164	032360	016777	.WORD .S.	
13165	032362		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.	
13166		000034	MICPC=MICPC+1	
13167	032362	016777	.WORD .S.	
13168	032364		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
13169		000035	MICPC=MICPC+1	
13170	032364	016652	.WORD .S.	
13171	032366		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
13172		000036	MICPC=MICPC+1	
13173	032366	016525	.WORD .S.	
13174	032370		MOVE # 124, MEM MARINC	
13175		000037	MICPC=MICPC+1	
13176	032370	016524	.WORD .S.	
13177	032372		MOVE # 124, MEM MARINC ;RESULT WITH C BIT SET.	
13178		000040	MICPC=MICPC+1	
13179	032372	016524	.WORD .S.	
13180	032374		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.	
13181		000041	MICPC=MICPC+1	
13182	032374	016525	.WORD .S.	
13183	032376		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY	
13184		000042	MICPC=MICPC+1	
13185	032376	016652	.WORD .S.	
13186	032400		MOVE # 252, MEM MARINC	
13187		000043	MICPC=MICPC+1	
13188	032400	016652	.WORD .S.	
13189	032402		MOVE # 252, MEM MARINC ;RESULT WITH C BIT SET.	
13190		000044	MICPC=MICPC+1	
13191	032402	016652	.WORD .S.	
13192	032404		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
13193		000045	MICPC=MICPC+1	
13194	032404	016652	.WORD .S.	
13195	032406		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.	
13196		000046	MICPC=MICPC+1	
13197	032406	016652	.WORD .S.	
13198	032410		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.	
13199		000047	MICPC=MICPC+1	
13200	032410	016777	.WORD .S.	
13201	032412		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.	
13202		000050	MICPC=MICPC+1	
13203	032412	016777	.WORD .S.	
13204	032414		MOVE # 7, SPAD <7> ;SET THE COUNT.	
13205		000051	MICPC=MICPC+1	
13206	032414	003007	.WORD .S.	
13207				
13208	032416		MOVE # 0, MLR ;MAR+0.	
13209		000052	MICPC=MICPC+1	
13210	032416	010000	.WORD .S.	
13211				
13212	032420		2S: MOVE # 0, BREG ;	
13213		000053	MICPC=MICPC+1	
13214	032420	000400	.WORD .S.	
13215	032422		SADD SPAD <1>, BREG ;CLEAR C BIT.	
13216		000054	MICPC=MICPC+1	
13217	032422	060401	.WORD .S. DD	
13218	032424		MOVE MEM, SPAD <0> MARINC ;GET THE FIRST OPERAND.	
13219		000055	MICPC=MICPC+1	

13220	032424	057220	.WORD .S.
13221	032426		SUB2C MEM,SPAD <0>,BR.SP,MARINC ;SPAD <0>:=\$FUNC.
13222		000056	MICPC=MICPC+1
13223	032426	057760	.WORD .S.
13224	032430		SIFEQ MEM,SPAD <0> 3\$;BRANCH IF GOOD.
13225			
13226			
13227	032430		SUB2C SPAD <0>,MEM,NOP
13228		000057	MICPC=MICPC+1
13229	032430	040360	.WORD .S.
13230	032432		BZ 3\$
13231		000060	MICPC=MICPC+1
13232	032432	101473	.WORD .S.
13233	032434		MOVE MEM,OUT1 <4> ;GOOD DATA
13234		000061	MICPC=MICPC+1
13235	032434	041224	.WORD .S.
13236	032436		MOVE BREG,OUT1 <5> ;BAD DATA.
13237		000062	MICPC=MICPC+1
13238	032436	061225	.WORD .S.
13239	032440		MOVE # 15,BREG ;SET TYPE OF ERROR.
13240		000063	MICPC=MICPC+1
13241	032440	000415	.WORD .S.
13242	032442		MOVE BREG,OUT1 <3> ;SET TYPE OF ERROR.
13243		000064	MICPC=MICPC+1
13244	032442	061223	.WORD .S.
13245	032444		MOVE # 17,BREG ;LOAD FUNCTION CODE...
13246		000065	MICPC=MICPC+1
13247	032444	000417	.WORD .S.
13248	032446		MOVE BREG,OUT1 <CSR7> ;LOAD IT...
13249		000066	MICPC=MICPC+1
13250	032446	061227	.WORD .S.
13251	032450		CALL EROR ;ALU 2'S COMP SUB ERROR...
13252	032450		MOVE # <MICPC+3>,BREG
13253		000067	MICPC=MICPC+1
13254	032450	000471	.WORD .S.
13255	032452		SBR EROR
13256		000070	MICPC=MICPC+1
13257	032452	104400	.WORD .S.
13258	032454		MOVE SPAD <4>,MLR ;RESET DATA POINTER...
13259		000071	MICPC=MICPC+1
13260	032454	070204	.WORD .S.
13261	032456		SBR 2\$;LOOP ON ERROR...
13262		000072	MICPC=MICPC+1
13263	032456	100453	.WORD .S.
13264	032460		CALL SCP1
13265	032460		MOVE # <MICPC+3>,BREG
13266		000073	MICPC=MICPC+1
13267	032460	000475	.WORD .S.
13268	032462		SBR SCP1
13269		000074	MICPC=MICPC+1
13270	032462	104427	.WORD .S.
13271	032464		MOVE SPAD <4>,MLR ;
13272		000075	MICPC=MICPC+1
13273	032464	070204	.WORD .S.
13274	032466		SBR 2\$;SCOPE THE DATA....
13275		000076	MICPC=MICPC+1

3\$:

13276	032466	100453	
13277	032470		6S: .WORD .S.
13278		000077	MOVE SPAD <4>,MLR ;RESET DATA POINTER...
13279	032470	070204	MICPC=MICPC+1
13280	032472		.WORD .S.
13281		000100	MOVE MEM,SPAD <0>,MARINC ;GET FIRST OPRAND...
13282	032472	057220	MICPC=MICPC+1
13283	032474		.WORD .S.
13284		000101	MOVE # 377,BREG ;
13285	032474	000777	MICPC=MICPC+1
13286	032476		.WORD .S.
13287		000102	SADD SPAD <2>,BREG ;SET C BIT...
13288	032476	060402	MICPC=MICPC+1
13289	032500		.WORD .S. DO
13290		000103	SUB2C MEM,SPAD <0>,BR.SP,MARINC ;SP 0 & BR = 2'S COMP SUB
13291	032500	057760	MICPC=MICPC+1
13292	032502		.WORD .S.
13293		000104	MOVE MEM,SPAD <3>,MARINC ;DUMMY INSTR, TO MARINC.
13294	032502	057223	MICPC=MICPC+1
13295	032504		.WORD .S.
13296			SIFEQ MEM,SPAD <0> 9S ;BR IF GOOD...
13297			
13298	032504		
13299		000105	SUB2C SPAD <0>,MEM,NOP
13300	032504	040360	MICPC=MICPC+1
13301	032506		.WORD .S.
13302		000106	BZ ;
13303	032506	101521	MICPC=MICPC+1
13304	032510		.WORD .S.
13305		000107	MOVE MEM,OUT1 <CSR4> ;GOOD DATA.
13306	032510	041224	MICPC=MICPC+1
13307	032512		.WORD .S.
13308		000110	MOVE BREG,OUT1 <CSR5> ;BAD DATA.
13309	032512	061225	MICPC=MICPC+1
13310	032514		.WORD .S.
13311		000111	MOVE # 23,BREG ;ERROR TYPE...
13312	032514	000423	MICPC=MICPC+1
13313	032516		.WORD .S.
13314		000112	MOVE BREG,OUT1 <CSR3> ;
13315	032516	061223	MICPC=MICPC+1
13316	032520		.WORD .S.
13317		000113	MOVE # 17,BREG ;ALU FUNCTION CODE.
13318	032520	000417	MICPC=MICPC+1
13319	032522		.WORD .S.
13320		000114	MOVE BREG,OUT1 <CSR7> ;LOAD IT...
13321	032522	061227	MICPC=MICPC+1
13322	032524		.WORD .S.
13323	032524		CALL EROR ;REPORT ERROR...
13324		000115	MOVE # <MICPC+3>,BREG
13325	032524	000517	MICPC=MICPC+1
13326	032526		.WORD .S.
13327		000116	SBR EROR
13328	032526	104400	MICPC=MICPC+1
13329	032530		.WORD .S.
13330		000117	MOVE SPAD <4>,MLR ;RESTORE DATA POINTER.
13331	032530	070204	MICPC=MICPC+1
			.WORD .S.

```

13332 032532          SBR          65                ;LOOP ON ERROR...
13333          MICPC=MICPC+1
13334 032532 000120  .WORD          .S.
13335 032534 100477          CALL          SCP1                ;SCOPE THE ERROR...
13336 032534          MOVE          # <MICPC+3>,BREG
13337          MICPC=MICPC+1
13338 032534 000121  .WORD          .S.
13339 032536 000523          SBR          SCP1
13340          MICPC=MICPC+1
13341 032536 000122  .WORD          .S.
13342 032540 104427          MOVE          SPAD <4>,MLR        ;RESTORE DATA POINTER...
13343          MICPC=MICPC+1
13344 032540 000123  .WORD          .S.
13345 032542 070204          SBR          65                ;SCOPE THE DATA...
13346          MICPC=MICPC+1
13347 032542 000124  .WORD          .S.
13348 032544 100477          MOVE          # 4,BREG            ;UPDATE BACKGROUND POINTER.
13349          MICPC=MICPC+1
13350 032544 000404  .WORD          .S.
13351 032546          SADD          BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
13352 032546 077004  .WORD          .S, MARINC!.DO
13353 032550          SDEC          SPAD <7>                ;IS IT DONE??
13354          MICPC=MICPC+1
13355 032550 000126  .WORD          .S,!.DSP
13356 032552 063167          BZ          45                ;YES, SCOPE THE TEST.
13357          MICPC=MICPC+1
13358 032552 000127  .WORD          .S.
13359 032554 101532          SBR          25                ;DO, THE NEXT.
13360          MICPC=MICPC+1
13361 032554 000130  .WORD          .S.
13362 032556 100453          CALL          SCPE                ;SCOPE THE TEST...
13363 032556          MOVE          # <MICPC+3>,BREG
13364          MICPC=MICPC+1
13365 032556 000131  .WORD          .S.
13366 032560 000533          SBR          SCPE
13367          MICPC=MICPC+1
13368 032560 000132  .WORD          .S.
13369 032562 104454          SBR          15                ;DO THE NEXT ITERATION...
13370          MICPC=MICPC+1
13371 032562 000133  .WORD          .S.
13372 032564          SALUT1 0,<DEC A>,SDEC,-1,-1,376,376,-1,-1,376,376,124,124,251,251,124,124,251,251,<A-1>
13373 032564          SXZ
13374
13375
13376
13377          ;***** TEST 61 *****
13378          ;#ALU TEST
13379          ;#TEST OF ALU FUNCTION DEC A WITH C BIT CLEARED.
13380          ;#TEST OF ALU FUNCTION DEC A WITH C BIT SET.
13381          ;#ALU FUNCTION (A-1)
13382          ;#LOAD MAIN MEMORY 16 WORDS OF DATA.
13383 032564          ;#PERFORM THE FUNCTION, VERIFY THE RESULTS..
13384          ;:*****
13385
13386 032564          STSTN
13387          ; TEST 61

```

13388							
13389	032564	012737	000061	001202	TST61:	MOV	#61,STSTNM ; LOAD THE NO. OF THIS TEST
13390	032572	012737	033112	001442		MOV	#TST62,NEXT ; POINT TO THE START OF NEXT TEST.
13391							;R1 CONTAINS BASE KMC11 ADDRESS
13392	032600	004737	035536			JSR	PC,LDRWRT ;LOAD-VERIFY-WAIT.
13393	032604	032620				MCT61	
13394	032606	104022				ERROR	22 ;TIME OUT ERROR...
13395	032610	012706	001200			MOV	#STACK,SP ;RESET STACK...
13396	032614	000177	146622			JMP	@NEXT ;GO TO NEXT TEST...
13397	032620				MCT61:		
13398	032620				IS:	MOVE	# 0,MLR ;SET MAR+LO.
13399		000000				MICPC=MICPC+1	
13400	032620	010000				.WORD	.S.
13401	032622					MOVE	# 0,MPR ;SET MAR+HI.
13402		000001				MICPC=MICPC+1	
13403	032622	004000				.WORD	.S.
13404	032624					MOVE	# 0,BREG ;
13405		000002				MICPC=MICPC+1	
13406	032624	000400				.WORD	.S.
13407	032626					MOVE	BREG,SPAD <16> ;FOR RETURN ADDRESS...
13408		000003				MICPC=MICPC+1	
13409	032626	063236				.WORD	.S.
13410	032630					MOVE	BREG,SPAD <0> ;
13411		000004				MICPC=MICPC+1	
13412	032630	063220				.WORD	.S.
13413	032632					MOVE	BREG,SPAD <1> ;
13414		000005				MICPC=MICPC+1	
13415	032632	063221				.WORD	.S.
13416	032634					MOVE	BREG,SPAD <2> ;
13417		000006				MICPC=MICPC+1	
13418	032634	063222				.WORD	.S.
13419	032636					SDEC	SPAD <2> ;
13420		000007				MICPC=MICPC+1	
13421	032636	063162			.WORD	.S.!.DSP	
13422	032640					MOVE	BREG,SPAD <4> ;
13423		000010				MICPC=MICPC+1	
13424	032640	063224				.WORD	.S.
13425	032642					MOVE	# 0,MEM MARINC ;LOAD THE DATA IN MEMORY.
13426		000011				MICPC=MICPC+1	
13427	032642	016400				.WORD	.S.
13428	032644					MOVE	# 0,MEM MARINC ;LOAD THE DATA IN MEM
13429		000012				MICPC=MICPC+1	
13430	032644	016400				.WORD	.S.
13431	032646					MOVE	# -1,MEM MARINC
13432		000013				MICPC=MICPC+1	
13433	032646	016777				.WORD	.S.
13434	032650					MOVE	# -1,MEM MARINC ;RESULT WITH C BIT SET.
13435		000014				MICPC=MICPC+1	
13436	032650	016777				.WORD	.S.
13437	032652					MOVE	# -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
13438		000015				MICPC=MICPC+1	
13439	032652	016777				.WORD	.S.
13440	032654					MOVE	# 0,MEM MARINC ;LOAD THE DATA IN MEMORY.
13441		000016				MICPC=MICPC+1	
13442	032654	016400				.WORD	.S.
13443	032656					MOVE	# 376,MEM MARINC

13444		000017	MICPC=MICPC+1	
13445	032658	016776	.WORD .S	
13446	032660		MOVE # 376, MEM MARINC	;RESULT WITH C BIT SET.
13447		000020	MICPC=MICPC+1	
13448	032660	016776	.WORD .S	
13449	032662		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
13450		000021	MICPC=MICPC+1	
13451	032662	016400	.WORD .S	
13452	032664		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
13453		000022	MICPC=MICPC+1	
13454	032664	016777	.WORD .S	
13455	032666		MOVE # -1, MEM MARINC	
13456		000023	MICPC=MICPC+1	
13457	032666	016777	.WORD .S	
13458	032670		MOVE # -1, MEM MARINC	;RESULT WITH C BIT SET.
13459		000024	MICPC=MICPC+1	
13460	032670	016777	.WORD .S	
13461	032672		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY
13462		000025	MICPC=MICPC+1	
13463	032672	016777	.WORD .S	
13464	032674		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
13465		000026	MICPC=MICPC+1	
13466	032674	016777	.WORD .S	
13467	032676		MOVE # 376, MEM MARINC	
13468		000027	MICPC=MICPC+1	
13469	032676	016776	.WORD .S	
13470	032700		MOVE # 376, MEM MARINC	;RESULT WITH C BIT SET.
13471		000030	MICPC=MICPC+1	
13472	032700	016776	.WORD .S	
13473	032702		MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
13474		000031	MICPC=MICPC+1	
13475	032702	016525	.WORD .S	
13476	032704		MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
13477		000032	MICPC=MICPC+1	
13478	032704	016525	.WORD .S	
13479	032706		MOVE # 124, MEM MARINC	
13480		000033	MICPC=MICPC+1	
13481	032706	016524	.WORD .S	
13482	032710		MOVE # 124, MEM MARINC	;RESULT WITH C BIT SET.
13483		000034	MICPC=MICPC+1	
13484	032710	016524	.WORD .S	
13485	032712		MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY.
13486		000035	MICPC=MICPC+1	
13487	032712	016652	.WORD .S	
13488	032714		MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
13489		000036	MICPC=MICPC+1	
13490	032714	016525	.WORD .S	
13491	032716		MOVE # 251, MEM MARINC	
13492		000037	MICPC=MICPC+1	
13493	032716	016651	.WORD .S	
13494	032720		MOVE # 251, MEM MARINC	;RESULT WITH C BIT SET.
13495		000040	MICPC=MICPC+1	
13496	032720	016651	.WORD .S	
13497	032722		MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
13498		000041	MICPC=MICPC+1	
13499	032722	016525	.WORD .S	

13500	032724		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY
13501		000042	MICPC=MICPC+1
13502	032724	016652	.WORD .S.
13503	032726		MOVE # 124, MEM MARINC
13504		000043	MICPC=MICPC+1
13505	032726	016524	.WORD .S.
13506	032730		MOVE # 124, MEM MARINC ;RESULT WITH C BIT SET.
13507		000044	MICPC=MICPC+1
13508	032730	016524	.WORD .S.
13509	032732		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
13510		000045	MICPC=MICPC+1
13511	032732	016652	.WORD .S.
13512	032734		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
13513		000046	MICPC=MICPC+1
13514	032734	016652	.WORD .S.
13515	032736		MOVE # 251, MEM MARINC ;LOAD THE DATA IN MEMORY.
13516		000047	MICPC=MICPC+1
13517	032736	016651	.WORD .S.
13518	032740		MOVE # 251, MEM MARINC ;RESULT WITH C BIT SET.
13519		000050	MICPC=MICPC+1
13520	032740	016651	.WORD .S.
13521	032742		MOVE # 7 SPAD <7> ;SET THE COUNT.
13522		000051	MICPC=MICPC+1
13523	032742	003007	.WORD .S.
13524			
13525	032744		MOVE # 0, MLR ;MAR←0.
13526		000052	MICPC=MICPC+1
13527	032744	010000	.WORD .S.
13528			
13529	032746		25: MOVE # 0, BREG ;
13530		000053	MICPC=MICPC+1
13531	032746	000400	.WORD .S.
13532	032750		SADD SPAD <1>, BREG ;CLEAR C BIT.
13533		000054	MICPC=MICPC+1
13534	032750	060401	.WORD .S.!.DO
13535	032752		MOVE MEM SPAD <0> MARINC ;GET THE FIRST OPERAND.
13536		000055	MICPC=MICPC+1
13537	032752	057220	.WORD .S.
13538	032754		SDEC SPAD <0>, BR.SP, MARINC ;
13539		000056	MICPC=MICPC+1
13540	032754	077560	.WORD .S.
13541	032756		SIFEQ MEM, SPAD <0> 3\$;BRANCH IF GOOD.
13542			
13543			
13544	032756		SUB2C SPAD <0>, MEM, NOP
13545		000057	MICPC=MICPC+1
13546	032756	0403E0	.WORD .S.
13547	032760		BZ 3\$
13548		000060	MICPC=MICPC+1
13549	032760	101473	.WORD .S.
13550	032762		MOVE MEM, OUT1 <4> ;GOOD DATA
13551		000061	MICPC=MICPC+1
13552	032762	041224	.WORD .S.
13553	032764		MOVE BREG, OUT1 <5> ;BAD DATA.
13554		000062	MICPC=MICPC+1
13555	032764	061225	.WORD .S.

13556	032766	000063	MOVE	# 15, BREG	;SET TYPE OF ERROR.
13557		000415	NICPC=NICPC+1		
13558	032766		.WORD	.S.	
13559	032770		MOVE	BREG, OUT1 <3>	;SET TYPE OF ERROR.
13560		000064	NICPC=NICPC+1		
13561	032770	061223	.WORD	.S.	
13562	032772		MOVE	# 07, BREG	;LOAD FUNCTION CODE...
13563		000065	NICPC=NICPC+1		
13564	032772	000407	.WORD	.S.	
13565	032774		MOVE	BREG, OUT1 <CSR7>	;LOAD IT...
13566		000066	NICPC=NICPC+1		
13567	032774	061227	.WORD	.S.	
13568	032776		CALL	EROR	;ALU DEC A ERROR...
13569	032776		MOVE	# <NICPC+3>, BREG	
13570		000067	NICPC=NICPC+1		
13571	032776	000471	.WORD	.S.	
13572	033000		SBR	EROR	
13573		000070	NICPC=NICPC+1		
13574	033000	104400	.WORD	.S.	
13575	033002		MOVE	SPAD <4>, MLR	;RESET DATA POINTER...
13576		000071	NICPC=NICPC+1		
13577	033002	070204	.WORD	.S.	
13578	033004		SBR	25	;LOOP ON ERROR...
13579		000072	NICPC=NICPC+1		
13580	033004	100453	.WORD	.S.	
13581	033006		CALL	SCP1	
13582	033006		MOVE	# <NICPC+3>, BREG	
13583		000073	NICPC=NICPC+1		
13584	033006	000475	.WORD	.S.	
13585	033010		SBR	SCP1	
13586		000074	NICPC=NICPC+1		
13587	033010	104427	.WORD	.S.	
13588	033012		MOVE	SPAD <4>, MLR	
13589		000075	NICPC=NICPC+1		
13590	033012	070204	.WORD	.S.	
13591	033014		SBR	25	;SCOPE THE DATA....
13592		000076	NICPC=NICPC+1		
13593	033014	100453	.WORD	.S.	
13594	033016		MOVE	SPAD <4>, MLR	;RESET DATA POINTER...
13595		000077	NICPC=NICPC+1		
13596	033016	070204	.WORD	.S.	
13597	033020		MOVE	MEM, SPAD <0>, MARINC	;GET FIRST OPRAND...
13598		000100	NICPC=NICPC+1		
13599	033020	057220	.WORD	.S.	
13600	033022		MOVE	# 377, BREG	
13601		000101	NICPC=NICPC+1		
13602	033022	000777	.WORD	.S.	
13603	033024		SADD	SPAD <2>, BREG	;SET C BIT...
13604		000102	NICPC=NICPC+1		
13605	033024	060402	.WORD	.S.	
13606	033026		SDEC	SPAD <0>, BR.SP, MARINC	;SP 0 & BR = DEC A
13607		000103	NICPC=NICPC+1		
13608	033026	077560	.WORD	.S.	
13609	033030		MOVE	MEM, SPAD <3>, MARINC	;DUMMY INSTR, TO MARINC.
13610		000104	NICPC=NICPC+1		
13611	033030	057223	.WORD	.S.	

```

13612 033032          SIFE0  MEM,SPAD <0>    95      ;BR IF GOOD...
13613
13614
13615 033032          SUB2C  SPAD <0>,MEM,NOP
13616          000105      MICPC=MICPC+1
13617 033032          .WORD  .S.
13618 033034          BZ      95
13619          000106      MICPC=MICPC+1
13620 033034          101521      .WORD  .S.
13621 033036          MOVE   MEM,OUT1 <CSR4> ;GOOD DATA.
13622          000107      MICPC=MICPC+1
13623 033036          041224      .WORD  .S.
13624 033040          MOVE   BREG,OUT1 <CSR5>          ;BAD DATA.
13625          000110      MICPC=MICPC+1
13626 033040          061225      .WORD  .S.
13627 033042          MOVE   # 23,BREG          ;ERROR TYPE...
13628          000111      MICPC=MICPC+1
13629 033042          000423      .WORD  .S.
13630 033044          MOVE   BREG,OUT1 <CSR3>          ;
13631          000112      MICPC=MICPC+1
13632 033044          061223      .WORD  .S.
13633 033046          MOVE   # 07,BREG          ;ALU FUNCTION CODE.
13634          000113      MICPC=MICPC+1
13635 033046          000407      .WORD  .S.
13636 033050          MOVE   BREG,OUT1 <CSR7>          ;LOAD IT...
13637          000114      MICPC=MICPC+1
13638 033050          061227      .WORD  .S.
13639 033052          CALL   EROR ;REPORT ERROR...
13640 033052          MOVE   # <MICPC+3>,BREG
13641          000115      MICPC=MICPC+1
13642 033052          000517      .WORD  .S.
13643 033054          SBR   EROR
13644          000116      MICPC=MICPC+1
13645 033054          104400      .WORD  .S.
13646 033056          MOVE   SPAD <4>,MLR          ;RESTORE DATA POINTER.
13647          000117      MICPC=MICPC+1
13648 033056          070204      .WORD  .S.
13649 033060          SBR   #5          ;LOOP ON ERROR...
13650          000120      MICPC=MICPC+1
13651 033060          100477      .WORD  .S.
13652 033062          95:  CALL   SCP1 ;SCOPE THE ERROR...
13653 033062          MOVE   # <MICPC+3>,BREG
13654          000121      MICPC=MICPC+1
13655 033062          000523      .WORD  .S.
13656 033064          SBR   SCP1
13657          000122      MICPC=MICPC+1
13658 033064          104427      .WORD  .S.
13659 033066          MOVE   SPAD <4>,MLR          ;RESTORE DATA POINTER...
13660          000123      MICPC=MICPC+1
13661 033066          070204      .WORD  .S.
13662 033070          SBR   #5          ;SCOPE THE DATA...
13663          000124      MICPC=MICPC+1
13664 033070          100477      .WORD  .S.
13665 033072          MOVE   # 4,BREG          ;UPDATE BACKGROUND POINTER.
13666          000125      MICPC=MICPC+1
13667 033072          000404      .WORD  .S.
    
```

```

13668 033074          SADD BREG,SPAD(4),MARINC ;ALSO DATA POINTER.
13669 033074 077004   .WORD $,MARINC!.DO
13670 033076          SDEC SPAD(7) ;IS IT DONE??
13671          000126   MICPC=MICPC+1
13672 033076 063167   .WORD $.!.DSP
13673 033100          BZ 48 ;YES, SCOPE THE TEST.
13674          000127   MICPC=MICPC+1
13675 033100 101532   .WORD $.
13676 033102          SBR 28 ;DO, THE NEXT.
13677          000130   MICPC=MICPC+1
13678 033102 107453   .WORD $.
13679 033104          4S: CALL SCPE ;SCOPE THE TEST...
13680 033104          MOVE # <MICPC+3>,BREG
13681          000131   MICPC=MICPC+1
13682 033104 000533   .WORD $.
13683 033106          SBR SCPE
13684          000132   MICPC=MICPC+1
13685 033106 104454   .WORD $.
13686 033110          SBR 18 ;DO THE NEXT ITERATION...
13687          000133   MICPC=MICPC+1
13688 033110 100400   .WORD $.
13689 033112          SPGMCK
13690 033112          SXZ
13691
13692
13693          ;***** TEST 62 *****
13694          ;*TEST OF PROGRAM CLOCK BIT
13695          ;*DO A MASTER CLEAR, VERIFY THAT PROGRAM CLOCK IS SET
13696          ;*WRITE PROGRAM CLOCK BIT TO A ONE, VERIFY THAT IT CLEARS.
13697          ;*AND THEN SETS SOME TIME LATER.
13698 033112          SXZ
13699          ;:*****
13700
13701 033112          STSTN
13702          ; TEST 62
13703          -----
13704 033112 012737 000062 001202 TST62: MOV #62,STSTNM ; LOAD THE NO. OF THIS TEST
13705 033120 012737 033260 001442   MOV #TST63,NEXT ; POINT TO THE START OF NEXT TEST.
13706          ;R1 CONTAINS BASE KMC11 ADDRESS
13707 033126 004737 035536          JSR PC,LDVRWT ;LOAD-VERIFY-WAIT.
13708 033132 033146          MCT62
13709 033134 104022          ERROR 22 ;TIME OUT ERROR...
13710 033136 012706 001200   MOV #STACK,SP ;RESET STACK...
13711 033142 000177 146274   JMP #NEXT ;GO TO NEXT TEST...
13712 033146          MCT62:
13713 033146          1S: MOVE #0,BREG ;PREPARE
13714          000000   MICPC=MICPC+1
13715 033146 000400   .WORD $.
13716 033150          MOVE BREG,SPAD(2) ; FOR
13717          000001   MICPC=MICPC+1
13718 033150 063222   .WORD $.
13719 033152          MOVE BREG,SPAD(3) ; DELAY.
13720          000002   MICPC=MICPC+1
13721 033152 063223   .WORD $.
13722 033154          MOVE BREG,SPAD(4) ;
13723          000003   MICPC=MICPC+1

```

13724	033154	063224	.WORD .S.	
13725	033156		MOVE BREG SPAD <16>	;FOR RETURN ADDRESS PURPOSE...
13726		000004	MICPC=MICPC+1	
13727	033156	063236	.WORD .S.	
13728	033160		MOVE INP1 <CSR11>,BREG	;CHECK IF PGM. CLK BIT SET.
13729		000005	MICPC=MICPC+1	
13730	033160	120620	.WORD .S.	
13731	033162		BH 25	;YES, GO RESET IT.
13732		000006	MICPC=MICPC+1	
13733	033162	103014	.WORD .S.	
13734	033164		MOVE # 14 MEM	;ERROR PROGRAM CLOCK BIT IS NOT SET.
13735		000007	MICPC=MICPC+1	
13736	033164	002414	.WORD .S.	
13737	033166		MOVE MEM,OUT1 <CSR3>	;ERROR TYPE...
13738		000010	MICPC=MICPC+1	
13739	033166	041223	.WORD .S.	
13740	033170		CALL ERROR1	;SET TYPE OF ERROR
13741	033170		MOVE # <MICPC+3>,BREG	
13742		000011	MICPC=MICPC+1	
13743	033170	000413	.WORD .S.	
13744	033172		SBR ERROR1	
13745		000012	MICPC=MICPC+1	
13746	033172	104401	.WORD .S.	
13747	033174		SBR 15	;LOOP ON ERROR.
13748		000013	MICPC=MICPC+1	
13749	033174	100400	.WORD .S.	
13750	033176		MOVE # 20 BREG	;GET SET TO CLEAR
13751		000014	MICPC=MICPC+1	
13752	033176	000420	.WORD .S.	
13753	033200		MOVE BREG,OUT1 <CSR11>	;PROGRAM CLOCK BIT.
13754		000015	MICPC=MICPC+1	
13755	033200	061231	.WORD .S.	
13756	033202		MOVE INP1 <CSR11>,BREG	;SEE IF IT IS CLEARED.
13757		000016	MICPC=MICPC+1	
13758	033202	120620	.WORD .S.	
13759	033204		BH 25	;NO, REPORT ERROR.
13760		000017	MICPC=MICPC+1	
13761	033204	103021	.WORD .S.	
13762	033206		SBR 45	;GO SEE IF IT SETS AGAIN.
13763		000020	MICPC=MICPC+1	
13764	033206	100426	.WORD .S.	
13765	033210		MOVE # 14 MEM	;TYPE
13766	033210		MICPC=MICPC+1	
13767		000021	.WORD .S.	
13768	033210	002414	MOVE MEM,OUT1 <CSR3>	
13769	033212		MICPC=MICPC+1	
13770		000022	.WORD .S.	
13771	033212	041223	CALL ERROR1	;PGM. CLK BIT DOES NOT CLEAR.
13772	033214		MOVE # <MICPC+3>,BREG	
13773	033214		MICPC=MICPC+1	
13774		000023	.WORD .S.	
13775	033214	000425	SBR ERROR1	
13776	033216		MICPC=MICPC+1	
13777		000024	.WORD .S.	
13778	033216	104401	SBR 15	;START AGAIN...
13779	033220			

25:

35:

13780 000025
 13781 033220 100400
 13782 033222
 13783 000026
 13784 033222 120620
 13785 033224
 13786 000027
 13787 033224 103042
 13788 033226
 13789 000030
 13790 033226 063062
 13791 033230
 13792 000031
 13793 033230 063103
 13794 033232
 13795 000032
 13796 033232 060603
 13797 033234
 13798 000033
 13799 033234 103435
 13800 033236
 13801 000034
 13802 033236 100426
 13803 033240
 13804 033240
 13805 000035
 13806 033240 002414
 13807 033242
 13808 000036
 13809 033242 041223
 13810 033244
 13811 033244
 13812 000037
 13813 033244 000441
 13814 033246
 13815 000040
 13816 033246 104401
 13817 033250
 13818 000041
 13819 033250 100400
 13820 033252
 13821 033252
 13822 000042
 13823 033252 000444
 13824 033254
 13825 000043
 13826 033254 104454
 13827 033256
 13828 000044
 13829 033256 100400
 13830 033260
 13831 033260
 13832
 13833
 13834
 13835

```

MICPC=MICPC+1
.WORD .S.
4S: MOVE INP1 (CSR11),BREG ;IS IT SET AGAIN?
MICPC=MICPC+1
.WORD .S.
BB4 SS ;YES, IT'S OK.
MICPC=MICPC+1
.WORD .S.
$INC SPAD (2) ;IN DELAY!
.WORD .S.! .DSP
SADC SPAD (3) ;DELAY!
.WORD .S.! .DSP
MICPC=MICPC+1
MOVE SPAD (3),BREG ;IS DELAYED ENOUGH?
MICPC=MICPC+1
.WORD .S.
BB7 6S ;YES, REPORT ERROR.
.WORD .S.
SBR 4S ;CONTINUE DELAY LOOP.
MICPC=MICPC+1
.WORD .S.
6S: MOVE 8 14 MEM ;SET TYPE OF ERROR
MICPC=MICPC+1
.WORD .S.
MOVE MEM,OUT1 (CSR3) ;
MICPC=MICPC+1
.WORD .S.
CALL ERROR1 ;ERROR PGM. CLK NOT SET.
MOVE 8 (MICPC+3),BREG
MICPC=MICPC+1
.WORD .S.
SBR ERROR1
MICPC=MICPC+1
.WORD .S.
SBR 1S ;LOOP ON ERROR.
MICPC=MICPC+1
.WORD .S.
5S: CALL SCPE ;SCOPE THE TEST.
MOVE 8 (MICPC+3),BREG
MICPC=MICPC+1
.WORD .S.
SBR SCPE
MICPC=MICPC+1
.WORD .S.
SBR 1S ;DO NEXT PASS.
MICPC=MICPC+1
.WORD .S.

```

SNOISE
SXZ

***** TEST 63 *****
MICRO-PROCESSOR NOISE TEST.

```

13836                                     ;WRITE ALL ZERO'S THEN ALL ONE'S THEN A DATA PATTERN TO
13837                                     ;THE IBUS, IBUS, SP, & MAIN MEMORY
13838                                     ;THEN GO AND READ BACK THE DATA PATTERN
13839                                     ;TO VERIFY THAT READING AND WRITING OF OTHER
13840                                     ;LOCATIONS DID NOT CHANGE DATA.
13841 033260 SXZ
13842                                     ;:*****
13843
13844 033260 STSTN
13845                                     ; TEST 63
13846                                     -----
13847 033260 012737 000063 001202 TST63: MOV #63,STSTNM ; LOAD THE NO. OF THIS TEST
13848 033266 012737 034064 001442 MOV #TST64,NEXT ; POINT TO THE START OF NEXT TEST.
13849                                     ;RI CONTAINS BASE KMC11 ADDRESS
13850 033274 004737 035536 .TSR PC,LDRWRT ;LOAD-VERIFY-WAIT.
13851 033300 033314 MCT63
13852 033302 104022 ERROR 22 ;TIME OUT ERROR...
13853 033304 012706 001200 MOV #STACK,SP ;RESET STACK.
13854 033310 000177 146126 JMP @NEXT ;GO TO NEXT TEST...
13855 033314 MCT63:
13856 033314 18S: MOVE #0,BREG ;START WITH ZERO.
13857 000000 MICPC=MICPC+1
13858 033314 000400 .WORD .S.
13859 033316 MOVE BREG,SPAD <0> ;
13860 000001 MICPC=MICPC+1
13861 033316 063220 .WORD .S.
13862 033320 MOVE BREG,SPAD <1> ;
13863 000002 MICPC=MICPC+1
13864 033320 063221 .WORD .S.
13865 033322 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES...
13866 000003 MICPC=MICPC+1
13867 033322 063236 .WORD .S.
13868
13869 033324 18S: MOVE #0,MLR ;MAR+0.
13870 000004 MICPC=MICPC+1
13871 033324 010000 .WORD .S.
13872 033326 MOVE #0,MPR ;
13873 000005 MICPC=MICPC+1
13874 033326 004000 .WORD .S.
13875 033330 MOVE BREG,OUT1 <CSR0> ;WRITE IBUS* REGISTERS.
13876 000006 MICPC=MICPC+1
13877 033330 061220 .WORD .S.
13878 033332 MOVE BREG,OUT1 <CSR2> ;WRITE IBUS* REGISTERS.
13879 000007 MICPC=MICPC+1
13880 033332 061222 .WORD .S.
13881 033334 MOVE BREG,OUT1 <CSR3> ;WRITE IBUS* REGISTERS.
13882 000010 MICPC=MICPC+1
13883 033334 061223 .WORD .S.
13884 033336 MOVE BREG,OUT1 <CSR4> ;WRITE IBUS* REGISTERS.
13885 000011 MICPC=MICPC+1
13886 033336 061224 .WORD .S.
13887 033340 MOVE BREG,OUT1 <CSR5> ;WRITE IBUS* REGISTERS.
13888 000012 MICPC=MICPC+1
13889 033340 061225 .WORD .S.
13890
13891 033342 MOVE BREG,OUT1 <CSR6> ;WRITE IBUS* REGISTERS.

```


13892		000013	MICPC=MICPC+1	
13893	033342	061226	.WORD .S.	
13894	033344		MOVE BREG OUT1 (CSR7)	;WRITE IBUS* REGISTERS.
13895		000014	MICPC=MICPC+1	
13896	033344	061227	.WORD .S.	
13897			::*	
13898			::*	
13899	033346		MOVE BREG OUT0 (0)	;WRITE IBUS REGISTERS.
13900		000015	MICPC=MICPC+1	
13901	033346	062220	.WORD .S.	
13902	033350		MOVE BREG OUT0 (1)	;WRITE IBUS REGISTERS.
13903		000016	MICPC=MICPC+1	
13904	033350	062221	.WORD .S.	
13905	033352		MOVE BREG OUT0 (2)	;WRITE IBUS REGISTERS.
13906		000017	MICPC=MICPC+1	
13907	033352	062222	.WORD .S.	
13908	033354		MOVE BREG OUT0 (3)	;WRITE IBUS REGISTERS.
13909		000020	MICPC=MICPC+1	
13910	033354	062223	.WORD .S.	
13911	033356		MOVE BREG OUT0 (4)	;WRITE IBUS REGISTERS.
13912		000021	MICPC=MICPC+1	
13913	033356	062224	.WORD .S.	
13914	033360		MOVE BREG OUT0 (5)	;WRITE IBUS REGISTERS.
13915		000022	MICPC=MICPC+1	
13916	033360	062225	.WORD .S.	
13917	033362		MOVE BREG OUT0 (6)	;WRITE IBUS REGISTERS.
13918		000023	MICPC=MICPC+1	
13919	033362	062226	.WORD .S.	
13920	033364		MOVE BREG OUT0 (7)	;WRITE IBUS REGISTERS.
13921		000024	MICPC=MICPC+1	
13922	033364	062227	.WORD .S.	
13923			::*	
13924			::*	
13925			::*	
13926	033366		MOVE BREG SPAD (3)	;WRITE SPAD LOCATIONS.
13927		000025	MICPC=MICPC+1	;WRITE SPAD LOCATIONS.
13928	033366	063223	.WORD .S.	
13929	033370		MOVE BREG SPAD (4)	;WRITE SPAD LOCATIONS.
13930		000026	MICPC=MICPC+1	
13931	033370	063224	.WORD .S.	
13932	033372		MOVE BREG SPAD (5)	;WRITE SPAD LOCATIONS.
13933		000027	MICPC=MICPC+1	
13934	033372	063225	.WORD .S.	
13935	033374		MOVE BREG SPAD (6)	;WRITE SPAD LOCATIONS.
13936		000030	MICPC=MICPC+1	
13937	033374	063226	.WORD .S.	
13938	033376		MOVE BREG SPAD (7)	;WRITE SPAD LOCATIONS.
13939		000031	MICPC=MICPC+1	
13940	033376	063227	.WORD .S.	
13941	033400		MOVE BREG SPAD (10)	;WRITE SPAD LOCATIONS.
13942		000032	MICPC=MICPC+1	
13943	033400	063230	.WORD .S.	
13944	033402		MOVE BREG SPAD (11)	;WRITE SPAD LOCATIONS.
13945		000033	MICPC=MICPC+1	
13946	033402	063231	.WORD .S.	
13947	033404		MOVE BREG, SPAD (12)	;WRITE SPAD LOCATIONS.

13948		000034		MICPC=MICPC+1	
13949	033404	063232		.WORD .S.	
13950	033406			MOVE BREG, SPAD <13>	;WRITE SPAD LOCATIONS.
13951		000035		MICPC=MICPC+1	
13952	033406	063233		.WORD .S.	
13953	033410			MOVE BREG, SPAD <14>	;WRITE SPAD LOCATIONS.
13954		000036		MICPC=MICPC+1	
13955	033410	063234		.WORD .S.	
13956	033412			MOVE BREG, SPAD <15>	;WRITE SPAD LOCATIONS.
13957		000037		MICPC=MICPC+1	
13958	033412	063235		.WORD .S.	
13959					
13960			25:		
13961	033414			MOVE BREG, MEM MARINC	;WRITE MEMORY LOCATION.
13962		000040		MICPC=MICPC+1	
13963	033414	076620		.WORD .S.	
13964	033416			SINC SPAD <0>	;UPDATE COUNT.
13965		000041		MICPC=MICPC+1	
13966	033416	063060	.WORD	.S.!.DSP	
13967	033420			SADC SPAD <1>	;UPDATE COUNT.
13968		000042		MICPC=MICPC+1	
13969	033420	063101	.WORD	.S.!.DSP	
13970	033422			MOVE BREG, SPAD <2>	;IS IT DONE?
13971		000043		MICPC=MICPC+1	
13972	033422	063222		.WORD .S.	
13973	033424			MOVE #4, BREG	;IS IT DONE?
13974		000044		MICPC=MICPC+1	
13975	033424	000404		.WORD .S.	
13976	033426			SIFHIS BREG, SPAD <1> 3S	;NO, CONTINUE.
13977					
13978					
13979	033426			SUB2C SPAD <1>, BREG, NOP	
13980		000045		MICPC=MICPC+1	
13981	033426	060361		.WORD .S.	
13982	033430			BC 64S	
13983		000046		MICPC=MICPC+1	
13984	033430	101050		.WORD .S.	
13985	033432			SBR 3S	
13986		000047		MICPC=MICPC+1	
13987	033432	100452		.WORD .S.	
13988	033434		64S:		
13989	033434			MOVE SPAD <2>, BREG	
13990		000050		MICPC=MICPC+1	
13991	033434	060602		.WORD .S.	
13992	033436			SBR 4S	;YES, WRITE THE NEXT.
13993		000051		MICPC=MICPC+1	
13994	033436	100454		.WORD .S.	
13995	033440		3S:	MOVE SPAD <2>, BREG	;RESTORE BREG.
13996		000052		MICPC=MICPC+1	
13997	033440	060602		.WORD .S.	
13998	033442			SBR 2S	;WRITE NEXT MEMORY LOCATION
13999		000053		MICPC=MICPC+1	
14000	033442	100440		.WORD .S.	
14001	033444		4S:		
14002	033444			SDEC SPAD <2>	;WAS IT ZERO?
14003		000054		MICPC=MICPC+1	

```

14004 033444 063162 .WORD .S.!.DSP
14005 033446 BZ 5S ;YES, NOW LOAD 377.
14006 000055 MICPC=MICPC+1
14007 033446 101462 .WORD .S.
14008 033450 MOVE BREG SPAD <2> ;WAS IT 377?
14009 000056 MICPC=MICPC+1
14010 033450 063222 .WORD .S.
14011 033452 $INC SPAD <2> ;WAS IT 377?
14012 000057 MICPC=MICPC+1
14013 033452 063062 .WORD .S.!.DSP
14014 033454 BC 6S ;YES, NOW LOAD 252.
14015 000060 MICPC=MICPC+1
14016 033454 101064 .WORD .S.
14017 033456 SBR 9S ;DONE START READING.
14018 000061 MICPC=MICPC+1
14019 033456 100471 .WORD .S.
14020 033460 5S: MOVE # 377, BREG ;LOAD 377.
14021 000062 MICPC=MICPC+1
14022 033460 000777 .WORD .S.
14023 033462 SBR 7S ;
14024 000063 MICPC=MICPC+1
14025 033462 100465 .WORD .S.
14026 033464 6S: MOVE # 252, BREG ;LOAD 252
14027 000064 MICPC=MICPC+1
14028 033464 000652 .WORD .S.
14029 033466 7S: MOVE # 0, SPAD <0> ;RESET SPAD 0.
14030 033466 MICPC=MICPC+1
14031 000065 .WORD .S.
14032 033466 003000 MOVE # 1, SPAD <1> ;RESET SPAD 1.
14033 033470 MICPC=MICPC+1
14034 000066 .WORD .S.
14035 033470 003001 SDEC SPAD <1> ;RESET SPAD 1.
14036 033472 MICPC=MICPC+1
14037 000067 .WORD .S.!.DSP
14038 033472 063161 SBR 1S ;CONTINUE
14039 033474 MICPC=MICPC+1
14040 000070 .WORD .S.
14041 033474 100404 ;:
14042 ;:
14043 ;:
14044 ;:
14045 033476 9S: MOVE BREG SPAD <2> ;SET UP FOR CHECKING.
14046 000071 MICPC=MICPC+1
14047 033476 063222 .WORD .S.
14048 033500 MOVE INP1 <CSRO>, SPAD <1> ;GET IBUS* REGISTER
14049 000072 MICPC=MICPC+1
14050 033500 123001 .WORD .S.
14051 033502 CALL CMPRE ;VERIFY IT.
14052 033502 MOVE # <MICPC+3>, BREG
14053 000073 MICPC=MICPC+1
14054 033502 000475 .WORD .S.
14055 033504 SBR CMPRE
14056 000074 MICPC=MICPC+1
14057 033504 100650 .WORD .S.
14058 033506 MOVE INP1 <CSR2>, SPAD <1> ;READ NEXT IBUS* REG.
14059 000075 MICPC=MICPC+1

```

14060	033506	123041	.WORD .S.
14061	033510		CALL CMPRE ;VERIFY IT.
14062	033510		MOVE # <MICPC+3>,BREG
14063		000076	MICPC=MICPC+1
14064	033510	000500	.WORD .S.
14065	033512		SBR CMPRE
14066		000077	MICPC=MICPC+1
14067	033512	100650	.WORD .S.
14068	033514		MOVE INP1 <CSR3>,SPAD <1> ;READ NEXT IBUS* REG.
14069		000100	MICPC=MICPC+1
14070	033514	123061	.WORD .S.
14071	033516		CALL CMPRE ;VERIFY IT.
14072	033516		MOVE # <MICPC+3>,BREG
14073		000101	MICPC=MICPC+1
14074	033516	000503	.WORD .S.
14075	033520		SBR CMPRE
14076		000102	MICPC=MICPC+1
14077	033520	100650	.WORD .S.
14078	033522		MOVE INP1 <CSR4>,SPAD <1> ;READ NEXT IBUS* REG.
14079		000103	MICPC=MICPC+1
14080	033522	123101	.WORD .S.
14081	033524		CALL CMPRE ;VERIFY IT.
14082	033524		MOVE # <MICPC+3>,BREG
14083		000104	MICPC=MICPC+1
14084	033524	000506	.WORD .S.
14085	033526		SBR CMPRE
14086		000105	MICPC=MICPC+1
14087	033526	100650	.WORD .S.
14088	033530		MOVE INP1 <CSR5>,SPAD <1> ;READ NEXT IBUS* REG.
14089		000106	MICPC=MICPC+1
14090	033530	123121	.WORD .S.
14091	033532		CALL CMPRE ;VERIFY IT.
14092	033532		MOVE # <MICPC+3>,BREG
14093		000107	MICPC=MICPC+1
14094	033532	000511	.WORD .S.
14095	033534		SBR CMPRE
14096		000110	MICPC=MICPC+1
14097	033534	100650	.WORD .S.
14098	033536		MOVE INP1 <CSR6>,SPAD <1> ;READ NEXT IBUS* REG.
14099		000111	MICPC=MICPC+1
14100	033536	123141	.WORD .S.
14101	033540		CALL CMPRE ;VERIFY IT.
14102	033540		MOVE # <MICPC+3>,BREG
14103		000112	MICPC=MICPC+1
14104	033540	000514	.WORD .S.
14105	033542		SBR CMPRE
14106		000113	MICPC=MICPC+1
14107	033542	100650	.WORD .S.
14108	033544		MOVE INP1 <CSR7>,SPAD <1> ;READ NEXT IBUS* REG.
14109		000114	MICPC=MICPC+1
14110	033544	123161	.WORD .S.
14111	033546		CALL CMPRE ;VERIFY IT.
14112	033546		MOVE # <MICPC+3>,BREG
14113		000115	MICPC=MICPC+1
14114	033546	000517	.WORD .S.
14115	033550		SBR CMPRE

14116		000116	MICPC=MICPC+1
14117	033550	100650	.WORD .S.
14118			***
14119			***
14120	033552		MOVE INPO <0>,SPAD <1> ;GET IBUS REG.
14121		000117	MICPC=MICPC+1
14122	033552	023001	.WORD .S.
14123	033554		CALL CNPRE ;VERIFY IT.
14124	033554		MOVE # <MICPC+3>,BREG
14125		000120	MICPC=MICPC+1
14126	033554	000522	.WORD .S.
14127	033556		SBR CNPRE
14128		000121	MICPC=MICPC+1
14129	033556	100650	.WORD .S.
14130	033560		MOVE INPO <1>,SPAD <1> ;GET NEXT IBUS-REGISTER.
14131		000122	MICPC=MICPC+1
14132	033560	023021	.WORD .S.
14133	033562		CALL CNPRE ;VERIFY IT.
14134	033562		MOVE # <MICPC+3>,BREG
14135		000123	MICPC=MICPC+1
14136	033562	000525	.WORD .S.
14137	033564		SBR CNPRE
14138		000124	MICPC=MICPC+1
14139	033564	100650	.WORD .S.
14140	033566		MOVE INPO <2>,SPAD <1> ;GET NEXT IBUS+REG.
14141		000125	MICPC=MICPC+1
14142	033566	023041	.WORD .S.
14143	033570		CALL CNPRE ;VERIFY IT.
14144	033570		MOVE # <MICPC+3>,BREG
14145		000126	MICPC=MICPC+1
14146	033570	000530	.WORD .S.
14147	033572		SBR CNPRE
14148		000127	MICPC=MICPC+1
14149	033572	100650	.WORD .S.
14150	033574		MOVE INPO <3>,SPAD <1> ;GET NEXT IBUS+REG.
14151		000130	MICPC=MICPC+1
14152	033574	023061	.WORD .S.
14153	033576		CALL CNPRE ;VERIFY IT.
14154	033576		MOVE # <MICPC+3>,BREG
14155		000131	MICPC=MICPC+1
14156	033576	000533	.WORD .S.
14157	033600		SBR CNPRE
14158		000132	MICPC=MICPC+1
14159	033600	100650	.WORD .S.
14160	033602		MOVE INPO <4>,SPAD <1> ;GET NEXT IBUS+REG.
14161		000133	MICPC=MICPC+1
14162	033602	023101	.WORD .S.
14163	033604		CALL CNPRE ;VERIFY IT.
14164	033604		MOVE # <MICPC+3>,BREG
14165		000134	MICPC=MICPC+1
14166	033604	000536	.WORD .S.
14167	033606		SBR CNPRE
14168		000135	MICPC=MICPC+1
14169	033606	100650	.WORD .S.
14170	033610		MOVE INPO <5>,SPAD <1> ;GET NEXT IBU
14171		000136	MICPC=MICPC+1

14172	033610	023121	.WORD .S.
14173	033612		CALL CNPRE
14174	033612		MOVE # <MICPC+3>,BREG ;VERIFY IT.
14175		000137	MICPC=MICPC+1
14176	033612	000541	.WORD .S.
14177	033614		SBR CNPRE
14178		000140	MICPC=MICPC+1
14179	033614	100650	.WORD .S.
14180	033616		MOVE INPO <6>,SPAD <1> ;GET NEXT IBUS+REG.
14181		000141	MICPC=MICPC+1
14182	033616	023141	.WORD .S.
14183	033620		CALL CNPRE
14184	033620		MOVE # <MICPC+3>,BREG ;VERIFY IT
14185		000142	MICPC=MICPC+1
14186	033620	000544	.WORD .S.
14187	033622		SBR CNPRE
14188		000143	MICPC=MICPC+1
14189	033622	100650	.WORD .S.
14190	033624		MOVE INPO <7>,SPAD <1> ;GET NEXT IBUS+REG.
14191		000144	MICPC=MICPC+1
14192	033624	023161	.WORD .S.
14193	033626		CALL CNPRE
14194	033626		MOVE # <MICPC+3>,BREG ;VERIFY IT.
14195		000145	MICPC=MICPC+1
14196	033626	000547	.WORD .S.
14197	033630		SBR CNPRE
14198		000146	MICPC=MICPC+1
14199	033630	100650	.WORD .S.
14200			
14201			::*
14202			::*
14203	033632		::*
14204		000147	MOVE SPAD <3>,BREG ;GET SPAD LOCATION.
14205	033632	060603	MICPC=MICPC+1
14206	033634		.WORD .S.
14207		000150	MOVE BREG SPAD <1> ;
14208	033634	063221	MICPC=MICPC+1
14209	033636		.WORD .S.
14210	033636		CALL CNPRE
14211		000151	MOVE # <MICPC+3>,BREG ;VERIFY IT.
14212	033636	000553	MICPC=MICPC+1
14213	033640		.WORD .S.
14214		000152	SBR CNPRE
14215	033640	100650	MICPC=MICPC+1
14216	033642		.WORD .S.
14217		000153	MOVE SPAD <4>,BREG ;GET NEXT SPAD LOCATION.
14218	033642	060604	MICPC=MICPC+1
14219	033644		.WORD .S.
14220		000154	MOVE BREG SPAD <1> ;
14221	033644	063221	MICPC=MICPC+1
14222	033646		.WORD .S.
14223		000155	MOVE SPAD <5>,BREG ;GET NEXT SPAD LOCATION.
14224	033646	060605	MICPC=MICPC+1
14225	033650		.WORD .S.
14226		000156	MOVE BREG SPAD <1> ;
14227	033650	063221	MICPC=MICPC+1
			.WORD .S.

14228	033652		CALL	CMPRE	;VERIFY IT.
14229	033652		MOVE	# <MICPC+3>,BREG	
14230		000157	MICPC=	MICPC+1	
14231	033652	000561	.WORD	.S.	
14232	033654		SBR	CMPRE	
14233		000160	MICPC=	MICPC+1	
14234	033654	100650	.WORD	.S.	
14235	033656		MOVE	SPAD <6>,BREG	;GET NEXT SPAD LOCATION.
14236		000161	MICPC=	MICPC+1	
14237	033656	060606	.WORD	.S.	
14238	033660		MOVE	BREG,SPAD <1>	;
14239		000162	MICPC=	MICPC+1	
14240	033660	063221	.WORD	.S.	
14241	033662		CALL	CMPRE	;VERIFY IT.
14242	033662		MOVE	# <MICPC+3>,BREG	
14243		000163	MICPC=	MICPC+1	
14244	033662	000565	.WORD	.S.	
14245	033664		SBR	CMPRE	
14246		000164	MICPC=	MICPC+1	
14247	033664	100650	.WORD	.S.	
14248	033666		MOVE	SPAD <7>,BREG	;GET NEXT SPAD LOCATION.
14249		000165	MICPC=	MICPC+1	
14250	033666	060607	.WORD	.S.	
14251	033670		MOVE	BREG,SPAD <1>	;
14252		000166	MICPC=	MICPC+1	
14253	033670	063221	.WORD	.S.	
14254	033672		CALL	CMPRE	;VERIFY IT.
14255	033672		MOVE	# <MICPC+3>,BREG	
14256		000167	MICPC=	MICPC+1	
14257	033672	000571	.WORD	.S.	
14258	033674		SBR	CMPRE	
14259		000170	MICPC=	MICPC+1	
14260	033674	100650	.WORD	.S.	
14261	033676		MOVE	SPAD <10>,BREG	;GET NEXT SPAD LOCATION.
14262		000171	MICPC=	MICPC+1	
14263	033676	060610	.WORD	.S.	
14264	033700		MOVE	BREG,SPAD <2>	;
14265		000172	MICPC=	MICPC+1	
14266	033700	063222	.WORD	.S.	
14267	033702		CALL	CMPRE	;VERIFY IT.
14268	033702		MOVE	# <MICPC+3>,BREG	
14269		000173	MICPC=	MICPC+1	
14270	033702	000575	.WORD	.S.	
14271	033704		SBR	CMPRE	
14272		000174	MICPC=	MICPC+1	
14273	033704	100650	.WORD	.S.	
14274	033706		MOVE	SPAD <11>,BREG	;GET NEXT SPAD LOCATION.
14275		000175	MICPC=	MICPC+1	
14276	033706	060611	.WORD	.S.	
14277	033710		MOVE	BREG,SPAD <1>	;
14278		000176	MICPC=	MICPC+1	
14279	033710	063221	.WORD	.S.	
14280	033712		CALL	CMPRE	;VERIFY IT.
14281	033712		MOVE	# <MICPC+3>,BREG	
14282		000177	MICPC=	MICPC+1	
14283	033712	000601	.WORD	.S.	

14284	033714		SBR	CMPRE
14285		000200	MICPC=MICPC+1	
14286	033714	100650	.WORD	.S.
14287	033716		MOVE	SPAD <12>,BREG ;GET NEXT SPAD LOCATION.
14288		000201	MICPC=MICPC+1	
14289	033716	060612	.WORD	.S.
14290	033720		MOVE	BREG,SPAD <1> ;
14291		000202	MICPC=MICPC+1	
14292	033720	063221	.WORD	.S.
14293	033722		CALL	CMPRE ;VERIFY IT.
14294	033722		MOVE	# <MICPC+3>,BREG
14295		000203	MICPC=MICPC+1	
14296	033722	000605	.WORD	.S.
14297	033724		SBR	CMPRE
14298		000204	MICPC=MICPC+1	
14299	033724	100650	.WORD	.S.
14300	033726		MOVE	SPAD <13>,BREG ;GET NEXT SPAD LOCATION.
14301		000205	MICPC=MICPC+1	
14302	033726	060613	.WORD	.S.
14303	033730		MOVE	BREG,SPAD <1> ;
14304		000206	MICPC=MICPC+1	
14305	033730	063221	.WORD	.S.
14306	033732		CALL	CMPRE ;VERIFY IT.
14307	033732		MOVE	# <MICPC+3>,BREG
14308		000207	MICPC=MICPC+1	
14309	033732	000611	.WORD	.S.
14310	033734		SBR	CMPRE
14311		000210	MICPC=MICPC+1	
14312	033734	100650	.WORD	.S.
14313	033736		MOVE	SPAD <14>,BREG ;GET NEXT SPAD LOCATION.
14314		000211	MICPC=MICPC+1	
14315	033736	060614	.WORD	.S.
14316	033740		MOVE	BREG,SPAD <1> ;
14317		000212	MICPC=MICPC+1	
14318	033740	063221	.WORD	.S.
14319	033742		CALL	CMPRE ;VERIFY IT.
14320	033742		MOVE	# <MICPC+3>,BREG
14321		000213	MICPC=MICPC+1	
14322	033742	000615	.WORD	.S.
14323	033744		SBR	CMPRE
14324		000214	MICPC=MICPC+1	
14325	033744	100650	.WORD	.S.
14326	033746		MOVE	SPAD <15>,BREG ;GET NEXT SPAD LOCATION.
14327		000215	MICPC=MICPC+1	
14328	033746	060615	.WORD	.S.
14329	033750		MOVE	BREG,SPAD <1> ;
14330		000216	MICPC=MICPC+1	
14331	033750	063221	.WORD	.S.
14332	033752		CALL	CMPRE ;VERIFY IT.
14333	033752		MOVE	# <MICPC+3>,BREG
14334		000217	MICPC=MICPC+1	
14335	033752	000621	.WORD	.S.
14336	033754		SBR	CMPRE
14337		000220	MICPC=MICPC+1	
14338	033754	100650	.WORD	.S.
14339				


```

14340          ;:*
14341          ;:*
14342 033756    000221    MOVE    # 0,MLR          ;
14343          010000    MICPC=MICPC+1          ;
14344 033756    010000    .WORD    .S.
14345 033760    000222    MOVE    # 0,MPR          ;
14346          004000    MICPC=MICPC+1          ;
14347 033760    004000    .WORD    .S.
14348 033762    000223    MOVE    BREG,SPAD <0>   ;SET THE COUNTER.
14349          063220    MICPC=MICPC+1          ;
14350 033762    000224    .WORD    .S.
14351 033764    000224    MOVE    # 3,SPAD <3>   ;SET THE COUNTER.
14352          003003    MICPC=MICPC+1          ;
14353 033764    000225    .WORD    .S.
14354 033766    000225    MOVE    MEM,BREG,MARINC ;GET THE CONTENTS OF MEMORY LOCATION.
14355          054620    MICPC=MICPC+1          ;
14356 033766    054620    .WORD    .S.
14357 033770    000226    $IFEQ  BREG,SPAD <2> 15$ ;BRANCH IF GOOD.
14358
14359
14360 033770    000226    SUBRC  SPAD <2>,BREG,NOP
14361          060362    MICPC=MICPC+1          ;
14362 033770    060362    .WORD    .S.
14363 033772    000227    BZ     15$
14364          101637    MICPC=MICPC+1          ;
14365 033772    000227    .WORD    .S.
14366 033774    000230    MOVE    MEM,OUT1 <CSR4> ;GOOD DATA
14367          041224    MICPC=MICPC+1          ;
14368 033774    000231    .WORD    .S.
14369 033776    000231    MOVE    BREG,OUT1 <CSR5> ;BAD DATA
14370          061225    MICPC=MICPC+1          ;
14371 033776    000232    .WORD    .S.
14372 034000    000232    MOVE    # 20,BREG      ;
14373          000420    MICPC=MICPC+1          ;
14374 034000    000233    .WORD    .S.
14375 034002    000233    MOVE    BREG,OUT1 <CSR3> ;TYPE OF ERROR.
14376          061223    MICPC=MICPC+1          ;
14377 034002    000234    .WORD    .S.
14378 034004    000234    CALL   ERROR1          ;DATA ERROR.
14379 034004    000234    MOVE    # <MICPC+3>,BREG
14380          000636    MICPC=MICPC+1          ;
14381 034004    000235    .WORD    .S.
14382 034006    000235    SBR    ERROR1          ;
14383          104401    MICPC=MICPC+1          ;
14384 034006    000236    .WORD    .S.
14385 034010    000236    SBR    10$            ;LOOP ON ERROR.
14386          100400    MICPC=MICPC+1          ;
14387 034010    000237    .WORD    .S.
14388 034012    000237    $INC  SPAD <0>        ;COUNT BY ONE.
14389          063060    MICPC=MICPC+1          ;
14390 034012    000240    .WORD    .S.!.DSP
14391 034014    000240    BC     16$            ;COUNT BY ONE.
14392          101242    MICPC=MICPC+1          ;
14393 034014    000241    .WORD    .S.
14394 034016    000241    SBR    12$            ;DO THE NEXT.
14395          MICPC=MICPC+1

```

KMC11 ALU TESTS

14396 034016 100625
 14397 034020
 14398 000242
 14399 034020 063163
 14400 034022
 14401 000243
 14402 034022 101645
 14403 034024
 14404 000244
 14405 034024 100625
 14406 034026
 14407 034026
 14408 000245
 14409 034026 000647
 14410 034030
 14411 000246
 14412 034030 104454
 14413 034032
 14414 000247
 14415 034032 100400
 14416 034034
 14417 000250
 14418 034034 063237
 14419 034036
 14420 000251
 14421 034036 060601
 14422 034040
 14423
 14424
 14425 034040
 14426 000252
 14427 034040 060362
 14428 034042
 14429 000253
 14430 034042 101663
 14431 034044
 14432 000254
 14433 034044 061225
 14434 034046
 14435 000255
 14436 034046 060602
 14437 034050
 14438 000256
 14439 034050 061224
 14440 034052
 14441 000257
 14442 034052 000420
 14443 034054
 14444 000260
 14445 034054 061223
 14446 034056
 14447 034056
 14448 000261
 14449 034056 000663
 14450 034060
 14451 000262

```

168: .WORD $
      SDEC SPAD (3) ;IS IT DONE??
      MICPC=MICPC+1
.WORD $.!.DSP
      BZ 175 ;YES.SCOPE IT.
      MICPC=MICPC+1
      .WORD $
      SBR 125 ;ELSE, DO NEXT.
      MICPC=MICPC+1
175: .WORD $
      CALL SCOPE ;SCOPE THE TEST...
      MOVE # (MICPC+3),BREG
      MICPC=MICPC+1
      .WORD $
      SBR SCOPE
      MICPC=MICPC+1
      .WORD $
      SBR 185 ;DO THE NEXT ITERATION.
      MICPC=MICPC+1
CMPRE: .WORD $
        MOVE BREG SPAD (17) ;SAVE THE RETURN ADDRESS.
        MICPC=MICPC+1
        .WORD $
        MOVE SPAD (1),BREG ;GET FOUND IN BREG.
        MICPC=MICPC+1
        .WORD $
        SIFEQ BREG,SPAD (2) 33$ ;BRANCH IF GOOD.

SUB2C SPAD (2),BREG,NOP
      MICPC=MICPC+1
      .WORD $
      BZ 33$
      MICPC=MICPC+1
      .WORD $
      MOVE BREG OUT1 (CSRS) ;BAD DATA.
      MICPC=MICPC+1
      .WORD $
      MOVE SPAD (2),BREG ;
      MICPC=MICPC+1
      .WORD $
      MOVE BREG OUT1 (CSR4) ;GOOD DATA.
      MICPC=MICPC+1
      .WORD $
      MOVE # 20,BREG ;
      MICPC=MICPC+1
      .WORD $
      MOVE BREG OUT1 (CSR3) ;TYPE OF ERROR.
      MICPC=MICPC+1
      .WORD $
      CALL EROR1 ;REPORT ERROR
      MOVE # (MICPC+3),BREG
      MICPC=MICPC+1
      .WORD $
      SBR EROR1
      MICPC=MICPC+1

```

14452 034060 104401
 14453 034062
 14454 034062
 14455 000263
 14456 034062 160617
 14457 034064
 14458 034064
 14459
 14460
 14461
 14462
 14463
 14464
 14465 034064
 14466
 14467
 14468 034064
 14469
 14470
 14471 034064 012737 000064 001202
 14472 034072 012737 034314 001442
 14473
 14474 034100 004737 035536
 14475 034104 034120
 14476 034106 104022
 14477 034110 012706 001200
 14478 034114 000177 145322
 14479 034120
 14480 034120
 14481 000006
 14482 034120 000400
 14483 034122
 14484 000001
 14485 034122 010000
 14486 034124
 14487 000002
 14488 034124 004000
 14489 034126
 14490 000003
 14491 034126 061224
 14492 034130
 14493 000004
 14494 034130 061225
 14495 034132
 14496 000005
 14497 034132 062220
 14498 034134
 14499 000006
 14500 034134 062221
 14501 034136
 14502 000007
 14503 034136 062224
 14504 034140
 14505 000010
 14506 034140 062225
 14507 034142

33S: .WORD .S.
 SBR SPAD <17> PAGED
 MICPC=MICPC+1
 .WORD .S.
 SHLRSR
 SXZ

***** TEST 64 *****
 * HELL RAISER TEST...
 * ONLY TO TEST MPRAM CONTROL LOGIC...
 * NOT FOR MAINTENANCE PURPOSE...
 ;*****

SXZ

STSTN

; TEST 64

TST64:

MOV #64,STSTNM ; LOAD THE NO. OF THIS TEST
 MOV #TST65,NEXT ; POINT TO THE START OF NEXT TEST

; R1 CONTAINS BASE KMC11 ADDRESS
 ; LOAD-VERIFY-WAIT.

JSR

PC,LDRWT

MCT64

ERROR

22

; TIME OUT ERROR...

MOV

#STACK,SP

; RESET STACK...

JMP

#NEXT

; GO TO NEXT TEST...

MCT64:

IS:

MOVE #0,BREG

MICPC=MICPC+1

.WORD .S.

; CLEAR MAR.

MOVE #0,MLR

MICPC=MICPC+1

.WORD .S.

; CLEAR MAR.

MOVE #0,MPR

MICPC=MICPC+1

.WORD .S.

; CLEAR BSEL4.

MOVE BREG,OUT1 <CSR4>

MICPC=MICPC+1

.WORD .S.

; CLEAR BSEL5.

MOVE BREG,OUT0 <0>

MICPC=MICPC+1

.WORD .S.

; CLEAR IN DATA LB.

MOVE BREG,OUT0 <1>

MICPC=MICPC+1

.WORD .S.

; CLEAR IN DATA HB.

MOVE BREG,OUT0 <4>

MICPC=MICPC+1

.WORD .S.

; CLEAR IN BA LB.

MOVE BREG,OUT0 <5>

MICPC=MICPC+1

.WORD .S.

; CLEAR IN BA HB.

MOVE BREG,SPAD <16> ; RETURN ADDRESS PURPOSE.

14508		000011	MICPC=MICPC+1	
14509	034142	063236	.WORD .S.	
14510	034144		MOVE # <HELDAT&377>,BREG ;SET IN+BA LB.	
14511		000012	MICPC=MICPC+1	
14512	034144	000712	.WORD .S.	
14513	034146		MOVE BREG,OUTO <4> ;	
14514		000013	MICPC=MICPC+1	
14515	034146	062224	.WORD .S.	
14516	034150		MOVE # <HELDAT/400&377>,BREG ;SET IN+BA+HB.	
14517		000014	MICPC=MICPC+1	
14518	034150	000470	.WORD .S.	
14519	034152		MOVE BREG,OUTO <5> ;	
14520		000015	MICPC=MICPC+1	
14521	034152	062225	.WORD .S.	
14522	034154		MOVE # 022,MEM ;	
14523		000016	MICPC=MICPC+1	
14524	034154	002422	.WORD .S.	
14525	034156		MOVE MEM,OUT1 <CSR2> ;SET FOR HELL RAISER...	
14526		000017	MICPC=MICPC+1	
14527	034156	041222	.WORD .S.	
14528	034160		MOVE # 357,BREG ;GET THE MASK.	
14529		000020	MICPC=MICPC+1	
14530	034160	000757	.WORD .S.	
14531	034162		MOVE INP1 <CSR11>,SPAD <0> ;GET UPMS REGISTER.	
14532		000021	MICPC=MICPC+1	
14533	034162	123220	.WORD .S.	
14534	034164		AND BREG,SPAD <0>,SPAD <0> ;	
14535		000022	MICPC=MICPC+1	
14536	034164	063260	.WORD .S.	
14537	034166		MOVE # 300,BREG ;SET BR+RQ & VCTR:=XX4.	
14538		000023	MICPC=MICPC+1	
14539	034166	000700	.WORD .S.	
14540	034170		OR BREG,SPAD <0>,OUT1 <11> ;	
14541		000024	MICPC=MICPC+1	
14542	034170	061311	.WORD .S.	
14543	034172		25: MOVE INP1 <CSR11>,BREG ;IS BR+RQ GRANTED???	
14544		000025	MICPC=MICPC+1	
14545	034172	120620	.WORD .S.	
14546	034174		BB7 25 ;NO,SPIN ON IT.	
14547		000026	MICPC=MICPC+1	
14548	034174	103425	.WORD .S.	
14549	034176		MOVE # 0,BREG ;	
14550		000027	MICPC=MICPC+1	
14551	034176	000400	.WORD .S.	
14552	034200		MOVE BREG,OUT1 <CSR11> ;CLEARS VCTR:XX4.	
14553		000030	MICPC=MICPC+1	
14554	034200	061231	.WORD .S.	
14555	034202		35: MOVE INP1 <CSR2>,BREG ;IS POP11 ALL SET???	
14556		000031	MICPC=MICPC+1	
14557	034202	120440	.WORD .S.	
14558	034204		BB4 35 ;NO,WAIT FOR IT.	
14559		000032	MICPC=MICPC+1	
14560	034204	103031	.WORD .S.	
14561	034206		45: MOVE # 001,BREG ;SET NPR+RQ BIT.	
14562		000033	MICPC=MICPC+1	
14563	034206	000401	.WORD .S.	

14578	034220	102400	MOVE BREG, OUT1 <CSR10>	;
14579	034222		MICPC=MICPC+1	
14580			.WORD .S.	
14581	034222	123101	55: MOVE INP1 <CSR10>, BREG	; IS NPR DONE?
14582	034224		MICPC=MICPC+1	
14583			.WORD .S.	
14584	034224	123122	BBO 65	
14585	034226		MICPC=MICPC+1	
14586			.WORD .S.	
14587	034226	100451	35: MOVE INP1 <CSR2>, BREG	; IS CSR LOADED???
14588	034230		MICPC=MICPC+1	
14589			.WORD .S.	
14590	034230	120440	BBI 45	; NO WAIT FOR IT...
14591	034232		MICPC=MICPC+1	
14592			.WORD .S.	
14593	034232	102435	MOVE INP1 <CSR4>, SPAD <1>	;
14594	034234		MICPC=MICPC+1	
14595			.WORD .S.	
14596	034234	123101	MOVE INP1 <CSR5>, SPAD <2>	;
14597	034236		MICPC=MICPC+1	
14598			.WORD .S.	
14599	034236	123122	SBR 125	; GO CHECK DATA...
14600	034240		MICPC=MICPC+1	
14601			.WORD .S.	
14602	034240	100435	1: MOVE INP1 <CSR2>, BREG	; IS CSR LOADED???
14603	034242		MICPC=MICPC+1	
14604			.WORD .S.	
14605	034242	023000	BBI 55	; NO, CHECK NPR DONE...
14606	034244		MICPC=MICPC+1	
14607			.WORD .S.	
14608	034244	060601	MOVE INP1 <CSR4>, SPAD <1>	; TRANSFER TO SPAD 1.
14609	034246		MICPC=MICPC+1	
14610			.WORD .S.	
14611			MOVE INP1 <CSR5>, SPAD <2>	; TRANSFER TO SPAD 2.
14612	034246		MICPC=MICPC+1	
14613			.WORD .S.	
14614	034246	060361	SBR 55	; GO CHECK NPR.
14615	034250		MICPC=MICPC+1	
14616			.WORD .S.	
14617	034250	101456	125: MOVE INPD <0>, SPAD <0>	; GET NPR DATA.
14618	034252		MICPC=MICPC+1	
14619			.WORD .S.	
			MOVE SPAD <1>, BREG	; GET CSR DATA.
			MICPC=MICPC+1	
			.WORD .S.	
			SIFNE BREG, SPAD <1> 155	; BR IF NO COMPARE.
			SUB2C SPAD <1>, BREG, NOP	
			MICPC=MICPC+1	
			.WORD .S.	
			BZ 645	
			MICPC=MICPC+1	
			.WORD .S.	
			SBR 155	
			MICPC=MICPC+1	

14620	034252	100462		.WORD .S.	
14621	034254		645:		
14622	034254			MOVE INPO <1>,SPAD <0>	;GET NPR DATA.
14623		000056		MICPC=MICPC+1	
14624	034254	023020		.WORD .S.	
14625	034256			MOVE SPAD <2>,BREG	;GET CSR DATA.
14626		000057		MICPC=MICPC+1	
14627	034256	060602		.WORD .S.	
14628	034260			SIFEB BREG,SPAD <2> 165	;BRANCH IF GOOD.
14629					
14630					
14631	034260			SUB2C SPAD <2>,BREG,NOP	
14632		000060		MICPC=MICPC+1	
14633	034260	060362		.WORD .S.	
14634	034262			BZ 165	
14635		000061		MICPC=MICPC+1	
14636	034262	101472		.WORD .S.	
14637	034264		155:	MOVE BREG OUT1 <CSR4>	;CSR DATA.
14638		000062		MICPC=MICPC+1	
14639	034264	061224		.WORD .S.	
14640	034266			MOVE SPAD <0>,BREG	;NPR DATA.
14641		000063		MICPC=MICPC+1	
14642	034266	060600		.WORD .S.	
14643	034270			MOVE BREG OUT1 <CSR4>	
14644		000064		MICPC=MICPC+1	
14645	034270	061224		.WORD .S.	
14646	034272			MOVE 8 25 BREG	;ERROR TYPE.
14647		000065		MICPC=MICPC+1	
14648	034272	000425		.WORD .S.	
14649	034274			MOVE BREG OUT1 <CSR3>	
14650		000066		MICPC=MICPC+1	
14651	034274	061223		.WORD .S.	
14652	034276			CALL ERROR1	;REPORT ERROR.
14653	034276			MOVE 8 <MICPC+3>,BREG	
14654		000067		MICPC=MICPC+1	
14655	034276	000471		.WORD .S.	
14656	034300			SBR ERROR1	
14657		000070		MICPC=MICPC+1	
14658	034300	104401		.WORD .S.	
14659	034302			SBR 15	;LOOP ON ERROR.
14660		000071		MICPC=MICPC+1	
14661	034302	100400		.WORD .S.	
14662	034304		165:	CALL SCPE	;SCOPE THE TEST.
14663	034304			MOVE 8 <MICPC+3>,BREG	
14664		000072		MICPC=MICPC+1	
14665	034304	000474		.WORD .S.	
14666	034306			SBR SCPE	
14667		000073		MICPC=MICPC+1	
14668	034306	104454		.WORD .S.	
14669	034310			SBR 15	;YES DO NEXT PASS.
14670		000074		MICPC=MICPC+1	
14671	034310	100400		.WORD .S.	
14672	034312	125125			
14673	034314		HEL DAT:	125125	
14674	034314		SPWR1		
14675			\$XZ		

```

14676
14677
14678
14679
14680
14681
14682 034314 SXZ
14683
14684
14685
14686 034314 STSTN
14687
14688
14689 034314 012737 000065 001202 TST65:
14690 034322 012737 003662 001442
14691
14692 034330 004737 035536 JSR PC,LDRWT
14693 034334 034350 MCT65
14694 034336 104022 ERROR 22
14695 034340 012706 001200 MOV BSTACK,SP
14696 034344 000177 145072 JMP ANEXT
14697 034350 MCT65:
14698 034350 15: MOVE #0,BREG
14699 000000 MICPC=MICPC+1
14700 034350 000400 .WORD .S.
14701 034352 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS ...
14702 000001 MICPC=MICPC+1
14703 034352 063236 .WORD .S.
14704 034354 MOVE BREG,SPAD <1>
14705 000002 MICPC=MICPC+1
14706 034354 063221 .WORD .S.
14707 034356 MOVE BREG,SPAD <2>
14708 000003 MICPC=MICPC+1
14709 034356 063222 .WORD .S.
14710 034360 MOVE BREG,SPAD <3>
14711 000004 MICPC=MICPC+1
14712 034360 063223 .WORD .S.
14713 034362 MOVE #220,BREG ;GET READY TO SETUP POWER FAIL..
14714 000005 MICPC=MICPC+1
14715 034362 000620 .WORD .S.
14716 034364 MOVE INP1 <CSRD>,SPAD <0> ;GET CNTL/O REGISTER..
14717 000006 MICPC=MICPC+1
14718 034364 123000 .WORD .S.
14719 034366 OR BREG,SPAD <0>,OUT1 <CSRD> ;SET RDO, POWER FAIL BIT..
14720 000007 MICPC=MICPC+1
14721 034366 061300 .WORD .S.
14722 034370 MOVE #200,BREG
14723 000010 MICPC=MICPC+1
14724 034370 000600 .WORD .S.
14725 034372 MOVE INP1 <CSR11>,SPAD <0> ;GET UPMS REGISTER.
14726 000011 MICPC=MICPC+1
14727 034372 123220 .WORD .S.
14728 034374 OR BREG,SPAD <0>,OUT1 <CSR11> ;SET BR+REQ,VCTR:=XXD..
14729 000012 MICPC=MICPC+1
14730 034374 061311 .WORD .S.
14731 034376 25: MOVE INP1 <CSR11>,BREG ;IS REQ GRANTED??

```

```

;***** TEST 65 *****
;#FORCE POWER FAIL TEST.
;#SET FORCE POWER FAIL VERIFY THAT PROCESSOR TRAPS TO LOC 24.
;#GOING DOWN AND COMING UP. VERIFY ALSO THAT BUS INIT WAS
;#BLOCKED FROM GETTING TO KMC DURING THE POWER FAIL .
;*****

```

```

; TEST 65
-----
; LOAD THE NO. OF THIS TEST
; POINT TO THE END OF PASS HANDLER.
;R1 CONTAINS BASE KMC11 ADDRESS
;LOAD-VERIFY-WAIT.
;TIME OUT ERROR...
;RESET STACK...
;GO TO NEXT TEST...

```

;FOR RETURN ADDRESS ...

;GET READY TO SETUP POWER FAIL..

;GET CNTL/O REGISTER..

;SET RDO, POWER FAIL BIT..

;GET UPMS REGISTER.

;SET BR+REQ,VCTR:=XXD..

;IS REQ GRANTED??

```

14732          000013          MICPC=MICPC+1
14733 034376 120620          .WORD          .S.
14734 034400          BB7          25          ;NO, SPIN ON IT.
14735          000014          MICPC=MICPC+1
14736 034400 103413          .WORD          .S.
14737 034402          35: MOVE INP1 <CSRD>,BREG          ;IS INTERRUPT SERVICED??
14738          000015          MICPC=MICPC+1
14739 034402 120400          .WORD          .S.
14740 034404          BB4          35          ;NO, WAIT...
14741          000016          MICPC=MICPC+1
14742 034404 103015          .WORD          .S.
14743          ;:;*
14744          ;:;*
14745          ;:;*
14746 034406          MOVE          8 002,BREG          ;GET READY TO SET FORCE POWER FAIL.
14747          000017          MICPC=MICPC+1
14748 034406 000402          .WORD          .S.
14749 034410          MOVE INP1 <CSR11>,SPAD <0>          ;GET UPMS REGISTER...
14750          000020          MICPC=MICPC+1
14751 034410 123220          .WORD          .S.
14752 034412          OR          BREG SPAD <0>,OUT1 <CSR11>          ;SET AC LO..
14753          000021          MICPC=MICPC+1
14754 034412 061311          .WORD          .S.
14755 034414          45:
14756
14757 034414          MOVE INP1 <CSRD>,BREG          ;IS POWER FAIL SERVED??
14758          000022          MICPC=MICPC+1
14759 034414 120400          .WORD          .S.
14760 034416          BB7          65          ;NO, UPDATE DELAY COUNT.
14761          000023          MICPC=MICPC+1
14762 034416 103425          .WORD          .S.
14763 034420          SBR          95          ;YES, SCOPE THE TEST..
14764          000024          MICPC=MICPC+1
14765 034420 100432          .WORD          .S.
14766 034422          65: SINC SPAD <1>          ;UPDATE DELAY COUNT.
14767          000025          MICPC=MICPC+1
14768 034422 063061          .WORD          .S.!.DSP
14769 034424          SADC SPAD <2>          ;UPDATE DELAY COUNT.
14770          000026          MICPC=MICPC+1
14771 034424 063102          .WORD          .S.!.DSP
14772 034426          SADC SPAD <3>          ;UPDATE DELAY COUNT.
14773          000027          MICPC=MICPC+1
14774 034426 063103          .WORD          .S.!.DSP
14775 034430          BC          125          ;IS IT DELAYED ENOUGH?
14776          000030          MICPC=MICPC+1
14777 034430 101032          .WORD          .S.
14778 034432          SBR          45          ;NO, WAIT
14779          000031          MICPC=MICPC+1
14780 034432 100422          .WORD          .S.
14781 034434          125:
14782 034434          95:
14783 034434          MOVE INP1 <CSRD>,BREG          ;DUMMY WAIT LOOP
14784          000032          MICPC=MICPC+1
14785 034434 120400          .WORD          .S.
14786 034436          SBR          95          ;DO THE NEXT PASS.
14787          000033          MICPC=MICPC+1
    
```


14788 034436 100432
 14789 034440
 14790
 14791
 14792
 14793
 14794
 14795
 14796
 14797
 14798
 14799
 14800
 14801
 14802
 14803
 14804
 14805
 14806 034440
 14807 034440
 14808 034440
 14809 000400
 14810 034440 063076
 14811 034442
 14812 000401
 14813 034442 063076
 14814 034444
 14815 000402
 14816 034444 061226
 14817 034446
 14818 000403
 14819 034446 063237
 14820 034450
 14821 000404
 14822 034450 000601
 14823 034452
 14824 000405
 14825 034452 123000
 14826 034454
 14827 000406
 14828 034454 061300
 14829 034456
 14830 000407
 14831 034456 123220
 14832 034460
 14833 000410
 14834 034460 000757
 14835 034462
 14836 000411
 14837 034462 063260
 14838 034464
 14839 000412
 14840 034464 000600
 14841 034466
 14842 000413
 14843 034466 061311

TSTEN: .WORD .S.

:SUBROUTINES
:-----

.EVEN

```

*****
*
* ERROR REPORT ROUTINE. INTERRUPTS AT LOCATION XX0.
* 1) CSR4:=GOOD DATA      2) CSR5:=BAD DATA
* 3) CSR3:=ERROR TYPE     4) CSR6:=ERROR PC IN MICRO-CODE.
* 5) CSR7:=MISCELLANEOUS INFORMATION.
*
*****

```

```

SBEGIN ,377
SLOC 1000
ERORD: SINC SPAD (16) ;PREPARE FOR RETURN.
        MICPC=MICPC+1
        .WORD .S.!.DSP
ERORD1: SINC SPAD (1F)
        MICPC=MICPC+1
        .WORD .S.!.DSP
        MOVE BREG,OUT1 (CSR6) ;ERROR PC.
        MICPC=MICPC+1
        .WORD .S.
        MOVE BREG,SPAD (17) ;SAVE RETURN ADDRESS.
        MICPC=MICPC+1
        .WORD .S.
        MOVE #201,BREG ;SET UP RD+I & ERROR
        MICPC=MICPC+1
        .WORD .S.
        MOVE INP1 (CSR0),SPAD (0) ;GET THE BSELO.
        MICPC=MICPC+1
        .WORD .S.
        OR BREG,SPAD (0),OUT1 (CSR0) ;SET RD+I & ERROR IN SELO.
        MICPC=MICPC+1
        .WORD .S.
        MOVE INP1 (CSR11),SPAD (0) ;GET UPMS REGISTER.
        MICPC=MICPC+1
        .WORD .S.
        MOVE #357,BREG ;GET THE MASK FOR PGMCLK.
        MICPC=MICPC+1
        .WORD .S.
        AND BREG,SPAD (0),SPAD (0) ;MASK OUT PGMCLK BIT.
        MICPC=MICPC+1
        .WORD .S.
        MOVE #200,BREG ;PREPARE FOR INTERRUPT.
        MICPC=MICPC+1
        .WORD .S.
        OR BREG,SPAD (0),OUT1 (CSR11) ;SET BR+RO AT XX0.
        MICPC=MICPC+1
        .WORD .S.

```

14844 034470
 14845 000414
 14846 034470 120620
 14847 034472
 14848 000415
 14849 034472 107414
 14850 034474
 14851 000416
 14852 034474 120400
 14853 034476
 14854 000417
 14855 034476 106016
 14856 034500
 14857 000420
 14858 034500 107423
 14859 034502
 14860 000421
 14861 034502 060616
 14862 034504
 14863 000422
 14864 034504 063017
 14865 034506
 14866 000423
 14867 034506 000400
 14868 034510
 14869 000424
 14870 034510 063236
 14871 034512
 14872 000425
 14873 034512 061220
 14874 034514
 14875 000426
 14876 034514 160617
 14877
 14878
 14879
 14880
 14881
 14882
 14883
 14884 034516
 14885 000427
 14886 034516 063076
 14887 034520
 14888 000430
 14889 034520 063076
 14890 034522
 14891 000431
 14892 034522 063237
 14893 034524
 14894 000432
 14895 034524 000602
 14896 034526
 14897 000433
 14898 034526 061220
 14899 034530

```

51S:  MOVE  INP1 <CSR11>,BREG      ;IS BR+RO GRANTED?
      MICPC=MICPC+1
      .WORD  $
      BBT  $16
      MICPC=MICPC+1                ;NO, SPIN ON IT
      .WORD  $

52S:  MOVE  INP1 <CSRO>,BREG      ;IS ERROR INTERRUPT SERVICED?
      MICPC=MICPC+1
      .WORD  $
      BBT  $26
      MICPC=MICPC+1                ;NO, SPIN ON IT.
      .WORD  $
      BBT  $36
      MICPC=MICPC+1                ;LOOP ON ERROR?
      .WORD  $
      MOVE  SPAD <16>,BREG      ;NO, POP THE RETURN ADDRESS.
      MICPC=MICPC+1
      .WORD  $
      SADD  BREG,SPAD <17>,SPAD <17> ;NO, POP THE RETURN ADDRESS.
      MICPC=MICPC+1
      .WORD  $

53S:  MOVE  # 0,BREG
      MICPC=MICPC+1
      .WORD  $
      MOVE  BREG,SPAD <16> ;RESET SPAD <16>...
      MICPC=MICPC+1
      .WORD  $
      MOVE  BREG,OUT1 <CSRO>    ;RESET THE TALK REGISTER...
      MICPC=MICPC+1
      .WORD  $
      SBR   SPAD <17> PAGED ;RETURN TO CALLER.
      MICPC=MICPC+1
      .WORD  $.

*****
*
* SCOPE DATA ROUTINE
*
*****
SCP10: SINC  SPAD <16>          ;PREPARE FOR RETURN.
      MICPC=MICPC+1
      .WORD  $.!DSP
SCP110: SINC  SPAD <16>          ;PREPARE FOR RETURN.
      MICPC=MICPC+1
      .WORD  $.!DSP
      MOVE  BREG,SPAD <17>    ;RETURN ADDRESS.
      MICPC=MICPC+1
      .WORD  $
      MOVE  # 202,BREG        ;BREG+RD+I, SCOPI.
      MICPC=MICPC+1
      .WORD  $
      MOVE  BREG,OUT1 <CSRO>  ;SET RD+I, SCOPI.
      MICPC=MICPC+1
      .WORD  $
      MOVE  INP1 <CSR11>,SPAD <0> ;GET UPMS REGISTER
  
```

```

14900      000434      MICPC=MICPC+1
14901      034530      123220      .WORD      .S.
14902      034532      MOVE      # 357,BREG      ;GET MASK FOR PGMCLK.
14903      000435      MICPC=MICPC+1
14904      034532      000757      .WORD      .S.
14905      034534      AND      BREG,SPAD <0>,SPAD <0> ;MASK OUT PGMCLK BIT.
14906      000436      MICPC=MICPC+1
14907      034534      063260      .WORD      .S.
14908      034536      MOVE      # 200,BREG      ;PREPARE FOR INTERRUPT.
14909      000437      MICPC=MICPC+1
14910      034536      000600      .WORD      .S.
14911      034540      OR      BREG,SPAD <0>,OUT1 <CSR11> ;SET BR+RD AT XXD.
14912      000440      MICPC=MICPC+1
14913      034540      061311      .WORD      .S.
14914      034542      54$: MOVE      INP1 <CSR11>,BREG      ;IS BR+RD GRANTED?
14915      000441      MICPC=MICPC+1
14916      034542      120620      .WORD      .S.
14917      034544      BB7      54$      ;NO, SPIN ON IT.
14918      000442      MICPC=MICPC+1
14919      034544      107441      .WORD      .S.
14920      034546      55$: MOVE      INP1 <CSRD>,BREG      ;IS SCOP1 SERVICED?
14921      000443      MICPC=MICPC+1
14922      034546      120400      .WORD      .S.
14923      034550      BB1      55$      ;NO, SPIN ON IT.
14924      000444      MICPC=MICPC+1
14925      034550      106443      .WORD      .S.
14926      034552      BB7      55$      ;LOOP ON ERROR?
14927      000445      MICPC=MICPC+1
14928      034552      107450      .WORD      .S.
14929      034554      MOVE      SPAD <16>,BREG ;NO, POP THE RETURN ADDRESS.
14930      000446      MICPC=MICPC+1
14931      034554      060616      .WORD      .S.
14932      034556      SADD     BREG,SPAD <17>,SPAD <17> ;NO, POP THE RETURN ADDRESS.
14933      000447      MICPC=MICPC+1
14934      034556      063017      .WORD      .S.
14935      034560      56$: MOVE      # 0,BREG      ;NO, POP THE RETURN ADDRESS.
14936      000450      MICPC=MICPC+1
14937      034560      000400      .WORD      .S.
14938      034562      MOVE     BREG,SPAD <16> ;RESET SPAD <16>...
14939      000451      MICPC=MICPC+1
14940      034562      063236      .WORD      .S.
14941      034564      MOVE     BREG,OUT1 <CSRD> ;RESET THE TALK REGISTER.
14942      000452      MICPC=MICPC+1
14943      034564      061220      .WORD      .S.
14944      034566      SBR      SPAD <17> PAGED ;RETURN TO CALLER.
14945      000453      MICPC=MICPC+1
14946      034566      160617      .WORD      .S.
14947
14948
14949
14950
14951
14952
14953
14954      034570
14955      000454

```

```

:*****
:
: SCOPE THE TEST ROUTINE.
:
:*****
SCPED: MOVE     BREG,SPAD <17> ;SAVE RETURN ADDRESS.
        MICPC=MICPC+1

```

```

14956 034570 063237 .WORD .S.
14957 034572 MOVE # 200,BREG ;RD+0, XN BREG
14958 000455 MICPC=MICPC+1
14959 034572 000600 .WORD .S.
14960 034574 MOVE BREG,OUT1 <CSR2> ;BET RD+0 IN CSR2.
14961 000456 MICPC=MICPC+1
14962 034574 061222 .WORD .S.
14963 034576 MOVE INP1 <CSR11>,SPAD <0> ;GET UPMS REG.
14964 000457 MICPC=MICPC+1
14965 034576 123220 .WORD .S.
14966 034600 MOVE # 357,BREG ;GET MASK FOR PGMCLK.
14967 000460 MICPC=MICPC+1
14968 034600 000757 .WORD .S.
14969 034602 AND BREG,SPAD <0>,SPAD <0> ;MASK OUT PGMCLK BIT.
14970 000461 MICPC=MICPC+1
14971 034602 063260 .WORD .S.
14972 034604 MOVE # 300,BREG ;SET BR R0 & VCTR:=XX4.
14973 000462 MICPC=MICPC+1
14974 034604 000700 .WORD .S.
14975 034606 OR BREG,SPAD <0>,OUT1 <CSR11> ;SET BR+RD & VCTR:=XX4.
14976 000463 MICPC=MICPC+1
14977 034606 061311 .WORD .S.
14978 034610 42S: MOVE INP1 <CSR11>,BREG ;IS BR+RD GRANTED?
14979 000464 MICPC=MICPC+1
14980 034610 120620 .WORD .S.
14981 034612 BB7 42S ;NO, SPIN ON IT.
14982 000465 MICPC=MICPC+1
14983 034612 107464 .WORD .S.
14984 034614 MOVE # 00,BREG ;PREPARE TO CLEAR VCTR:=XX4.
14985 000466 MICPC=MICPC+1
14986 034614 000400 .WORD .S.
14987 034616 MOVE INP1 <CSR11>,SPAD <0> ;GET UPMS REG
14988 000467 MICPC=MICPC+1
14989 034616 123220 .WORD .S.
14990 034620 AND BREG,SPAD <0>,OUT1 <CSR11> ;RESET XX4
14991 000470 MICPC=MICPC+1
14992 034620 061271 .WORD .S.
14993 034622 45S: MOVE INP1 <CSR2>,BREG ;IS INTERRUPT SERVICED.
14994 000471 MICPC=MICPC+1
14995 034622 120440 .WORD .S.
14996 034624 BB7 45S ;NO, SPIN ON IT.
14997 000472 MICPC=MICPC+1
14998 034624 107471 .WORD .S.
14999 034626 SBR SPAD <17> PAGED ;RETURN
15000 000473 MICPC=MICPC+1
15001 034626 160617 .WORD .S.
15002
15003 *****
15004 *
15005 * MICRO-PROCESSOR PC SEQUENCE ERROR.
15006 *
15007 *****
15008
15009 UPCSQR: MOVE # 24,BREG ;ERROR CODE.
15010 000474 MICPC=MICPC+1
15011 034630 000424 .WORD .S.

```

```

15012 034632 MOVE BREG,OUT1 (CSR3) ;SET ERROR TYPE.
15013 000475 MICPC=MICPC+1
15014 034632 061223 .WORD .S.
15015 034634 MOVE #201,BREG ;SET UP FOR ERROR REPORT.
15016 000476 MICPC=MICPC+1
15017 034634 000601 .WORD .S.
15018 034636 MOVE BREG,OUT1 (CSR0) ;
15019 000477 MICPC=MICPC+1
15020 034636 061220 .WORD .S.
15021 034640 MOVE #200,BREG ;BREG+BR-RQ. AT VCTR:=XX0.
15022 000500 MICPC=MICPC+1
15023 034640 000600 .WORD .S.
15024 034642 MOVE BREG,CJT1 (CSR11) ;IS INTERRUPT SERVICED??
15025 000501 MICPC=MICPC+1
15026 034642 061231 .WORD .S.
15027 034644 15: MOVE INP1 (CSR11),BREG ;IS INTERRUPT SERVICED??
15028 000502 MICPC=MICPC+1
15029 034644 120620 .WORD .S.
15030 034646 BB7 IS ;SPIN ON IT.
15031 000503 MICPC=MICPC+1
15032 034646 107502 .WORD .S.
15033 034650 25: MOVE INP1 (CSR0),BREG ;IS ERROR SERVICED??
15034 000504 MICPC=MICPC+1
15035 034650 120400 .WORD .S.
15036 034652 BB0 AS ;WAIT...
15037 000505 MICPC=MICPC+1
15038 034652 106104 .WORD .S.
15039 034654 CALL SCPE0 ;SCOPE THE FAILING TEST.
15040 034654 MOVE # (MICPC+3),BREG
15041 000506 MICPC=MICPC+1
15042 034654 000510 .WORD .S.
15043 034656 SBR SCPE0
15044 000507 MICPC=MICPC+1
15045 034656 104454 .WORD .S.
15046 034660 35: MOVE INP1 (CSR0),BREG ;WAIT FOR PDP11 TO
15047 000510 MICPC=MICPC+1
15048 034660 120400 .WORD .S.
15049 034662 SBR AS ;.....
15050 000511 MICPC=MICPC+1
15051 034662 104510 .WORD .S.
15052 034664

```

MCSREN:

```

*****
*
* INPUT INTERRUPT SERVICE ROUTINE
* 1. TAKES CARE OF ERROR REPORTS.
* 2. PROVIDE SERVICE FOR SCOPE DATA.
*
*****

```

```

15063 034664 013737 001442 035356 INISR: :MOV R0,-(SP) ;PUSH R0.
15064 034672 013737 001206 035354 :MOV NEXT,SYNXT ;SAVE NEXT TEST ADDRESS.
15065 034700 032711 000001 :MOV SLPADR,SVLPDR ;SAVE LOOP ADDRESS.
15066 034704 001012 :BIT #BIT0,(R1) ;IS ERROR BIT SET?
15067 034706 032711 000002 :BNE ERR1 ;YES, SERVE THE ERROR INTERRUPT.
:BIT #BIT1,(R1) ;IS IT SCOPE DATA SET?

```

P46-

```

15068 034712 001075      BNE      SCPDT      ;YES, GO SERVE IT.
15069 034714 032711 000020    BIT      #BIT4, (R1) ;IS PWR. FAIL SERVICE SET?
15070 034720 001105      BNE      PWFAIL     ;GO SERVE POWER FAIL INTERRUPT.
15071
15072 034728 104401 011032    TYPE    ,NILINT     ;TYPE ILLEGAL INTERRUPT.
15073 034728 000000      HALT                    ;WAIT FOR OPERATOR ACTION.
15074 034730 000577      BR      RETN1        ;CONTINUE.
15075 034732 012737 035022 001442  ERR1:  MOV     #35, NEXT    ;LOAD RETURN ADDRESS.
15076 034734 012737 035022 001206      MOV     #35, SLPADR   ;LOAD RETURN ADDRESS.
15077 034736 116137 000004 001224      MOVVB  4(R1), SCNDAT  ;LOAD GOOD DATA.
15078 034734 116137 000005 001225      MOVVB  5(R1),        ;LOAD BAD DATA.
15079 034732 116137 000006 001222      MOVVB  6(R1),        ;LOAD UP PC.
15080 034770 042737 000777 035020      BIC     #77          ;CLEAR THE ERROR TYPE FIELD.
15081 034776 156137 000003 035020      BISB   3(R1), #5     ;SET THAT IN ERROR TRAP.
15082 035004 116137 000007 001220      MOVVB  7(R1), SCDADR ;SET CSR FOR REFERENCE.
15083 035012 012737 035354 007070      MOV     #SVLPDR, ERTAB1 ;SAVE ERROR PC...
15084
15085 035020 104000      65:    ERROR
15086 035022 012737 001460 007070 35:    MOV     #SAVPC, ERTAB1 ;REPORT THE ERROR.
15087 035030 032777 000400 144202      BIT     #SM08, #SMR   ;RESTORE ERROR PC POINTER.
15088 035036 001020      BNE     15          ;GO TO TOP OF TEST?
15089 035040 032777 002000 144172      BIT     #SM10, #SMR   ;BRANCH IF YES.
15090 035046 001412      BEQ    25          ;GO TO NEXT TEST?
15091 035050 013737 035356 001206      MOV     SVNXT, SLPADR ;BRANCH IF NO.
15092 035052 012705 001200      MOV     #STACK, SP   ;SET FOR NEXT TEST.
15093 035054 012737 000200 177776      MOV     #PR4, PS     ;RESET SP.
15094 035070 000177 144112      JMP     #SLPADR      ;SET PRIORITY 4 SO KMC CAN INTRRUPT.
15095 035074 042711 000200 25:    BIC     #BIT7, (R1)   ;GO TO SPECIFIED TEST.
15096 035100      15:    ;MOV     #1, CLKCTR   ;CLEAR LOOP ON ERROR BIT.
15097 035100 042711 000001      BIC     #BIT0, (R1)  ;RESET CLOCK.
15098 035104 000511      BR     RETN1        ;CLEAR ERROR BIT IN BSELO.
15099 035106 005037 001444      SCPT:  CLR     LOCK    ;RETURN.
15100 035112 104405      SCPT:  SCOP1        ;CLEAR FLAG
15101 035114 005737 001444      TST    LOCK         ;SCOPE THE DATA.
15102 035120 001002      BNE    SCPT1       ;IS FLAG SET?-I.E SCOPE THE DATA.
15103 035122 042711 000200      BIC     #BIT7, (R1)  ;YES, GO SET ACCORDINGLY.
15104 035124      SCPT1: BIC     #BIT1, (R1)  ;CLEAR SCOPE DATA LOOP BIT.
15105 035126 042711 000002      BR     RETN1        ;RETURN.
15106 035132 000476      ;:;: THIS IS PART OF POWER FAIL TEST
15107 ;:;: AFTER KMC CAUSES FORCE POWER FAIL
15108 ;:;: AND SETTING UP FOR POWER FAIL THIS
15109 ;:;: FUNCTIONS DONE HERE.
15110
15111 035134 012737 035202 001442  PWFAIL: MOV     #305, NEXT    ;RESTORE.
15112 035142 005037 011112      CLR     TEMP
15113 035146 012737 035202 001206      MOV     #305, SLPADR ;RESTORE.
15114 035154 013746 000024      MOV     #24, -(SP)   ;STORE POWER FAIL ADDRESS.
15115 035160 012737 035204 000024      MOV     #125, #24    ;SET UP FOR FORCE POWER FAIL.
15116 035166 042711 000020      BIC     #BIT4, (R1)  ;SIGNAL POWER FAIL SET UP.
15117 035172 005237 011112 155:  INC     TEMP        ;WAIT FOR POWER FAIL.
15118 035176 001375      BNE    155         ;BR IF DELAY NOT DONE.
15119 035200 104017      ERROR  17          ;ERROR, NO POWER FAIL.
15120 035202 000427      305:  BR     335
15121 ;:;:
15122 ;:;:
15123 ;:;:

```

RESTARTS HERE IN CASE OF POWER FAIL.


```

15180 035470 105061 000003 CLR      3(R1)      ;CLEAN THE WORLD...
15181 035474 013737 001206 001442 MOV      SVLPAD, NEXT ;ELSE, START FROM BEGINING.
15182 035502 005137 036146 COM      DONE
15183 035506 012737 000001 036150 RET1:   MOV      #1, CLKCTR ;RESET CLOCK.
15184 035514 000002 RTI      ;RETURN TO WAIT CLOCK.
15185 035516 012737 000001 036150 RET2:   MOV      #1, CLKCTR ;RESET WAIT CLOCK.
15186 035524 142761 000200 000002 BICB    #BIT7, 2(R1) ;CLEAR RD+0.
15187 035532 000002 RTI      ;RETURN.
15188 035534 000000 SVLPAD: 0

```

```

*****
ROUTINE TO LOAD MICRO-CODE INTO CRAM...
1. LOADS TEST.
2. LOADS UTILITY ROUTINES...
R1 CONTAINS THE CSR AT THE TIME OF ENTRY...
*****

```

```

15200 035536 000000
15201 035538 000000
15202 035540 000000
15203 035542 000000
15204 035544 000000
15205 035546 000000
15206 035548 000000
15207 035550 000000
15208 035552 000000
15209 035554 000000
15210 035556 000000
15211 035558 000000
15212 035560 000000
15213 035562 000000
15214 035564 000000
15215 035566 000000
15216 035568 000000
15217 035570 000000

```

```

LDVANT:
MCRAM:  ;CLR      RD
        MOV      #2000, RD ;RD=POINTS TO CRAM ADDRESS.
        MOV      #BIT14, (R1) ;MASTER CLEAR KMC11
        BIC      #BIT15:BIT14, (R1) ;AND SHUT IT OFF.
        MOV      #10474, R2 ;RD=UPCSR(MICRO-INSTRUCTION)
        CLR      (R1) ;START WITH CLEAN WORLD.
65:    MOV      RD, 4(R1) ;LOAD CRAM ADDRESS.
        MOV      R2, 6(R1) ;LOAD INSTRUCTION.
        MOV      #BIT10, (R1) ;SET RD+0.
        MOV      #BIT13:BIT10, (R1) ;WRITE IT.
        DEC      RD ;COUNT BY ONE.
        BNE     65 ;BR IF NOT DONE.
        MOV      2(SP)+, R2 ;SAVE START ADDRESS...
        MOV      R2, TSTRT ;POP RETURN ADDRESS...
        ADD     R2, -(SP)
        MOV      NEXT, MCEND ;SET UP END POINTER.
        CMP     #EOP, MCEND ;IS IT LAST TEST?
        BNE     15 ;NO, THEN IT'S O.K.
        MOV      #TSTRT, MCEND ;SET UP END POINTER.
15:    CLR      (R1) ;START WITH CLEAN WORLD.
        MOV      RD, 4(R1) ;LOAD CRAM ADDRESS.
        MOV      (R2)+, 6(R1) ;LOAD WORD TO BE WRITTEN.
        MOV      #BIT10, (R1) ;WRITE IT...
        MOV      #BIT13:BIT10, (R1) ;WRITE IT.
        INC     RD ;UPDATE CRAM ADDRESS.
        CMP     #2000, RD ;OVER FLOW??
        BLT     25 ;YES, THATS IT! NO MORE.
        CMP     MCEND, R2 ;IS IT DONE?
        BGT     15 ;NO, CONTINUE LOADING.
25:    CMP     #MCSREN, MCEND ;IS UTILITY ALSO LOADED?
        BEQ     35 ;YES, RETURN.

```


15236	035712	012700	000400	
15237	035716	012702	034440	
15238	035722	012737	034664	036154
15239	035730	000744		
15240	035732	005011		

```

MOV      #400,R0      ;R0=POINTS CRAM ADDRESS.
MOV      @R0,R2      ;R2=POINTS TO MICRO-CODE.
MOV      @R0,R2      ;SET UP END POINTER.
BR       IS          ;START LOADING UTILITY.
CLR      (R1)        ;END WITH CLEAN WORLD.

```

```

*****
*
*   VERIFY THE MICRO-CODE LOADED.
*   VERIFIES THE LOADED TEST MICRO-CODE. AND UTILITY
*   ROUTINES.
*   R1 CONTAINS DEVICE CSR AT THE TIME OF ENTRY...
*
*****

```

15241	035734	005002		
15242	035736	013700	036152	
15243	035742	013737	001442	036154
15244	035750	022737	003662	036154
15245	035756	001003		
15246	035760	012737	034440	036154
15247	035766	005011		
15248	035770	010261	000004	
15249	035774	012711	002000	
15250	035800	026120	000006	
15251	036004	001403		
15252	036006	104401	036156	
15253	036012	000000		
15254	036014	005202		
15255	036016	023700	036154	
15256	036022	003361		
15257	036024	022737	034664	036154
15258	036032	001410		
15259	036034	012702	000400	
15260	036040	012737	034664	036154
15261	036046	012700	034440	
15262	036052	000745		
15263	036054	005011		
15264	036056	012711	040000	
15265	036062	042711	140000	

```

VERIFY: CLR      R2      ;R2 GETS CRAM ADDRESS.
        MOV      TSTRT,R0 ;LOAD START ADDRESS...
        MOV      NEXT,MCEND ;SET UP END POINTER.
        CMP      @R0,MCEND ;IS IT LAST TEST??
        BNE     IS      ;NO, THEN IT'S O.K.
        MOV      @R0,MCEND ;SET UP END POINTER.
        CLR      (R1)     ;CLEAR SELD.
        MOV      R2,4(R1) ;SET THE CRAM ADDRESS.
        MOV      @BIT10,(R1) ;SET R0RD.
        CMP      6(R1),(R0)+ ;CHECK IF RIGHT?
        BEQ     JS      ;CHECK IF DONE?
        TYP     ,ALDER    ;LOADING ERROR.
        HA     ;WAIT FOR OPERATOR.
        INC     R2      ;BUMP CRAM ADDRESS.
        CMP     MCEND,R0 ;IS IT DONE??
        BGT     IS      ;NO, CHECK NEXT.
        CMP     @R0,MCEND ;IS IT DONE??
        BEQ     JS      ;YES RETURN...
        MOV     #400,R2  ;R2 CONTAINS CRAM ADDRESS...
        MOV     @R0,MCEND ;SET UP END POINTER...
        MOV     @R0,R0  ;R0=POINTS TO MICRO-CODE.
        BR      IS     ;START CHECKING UTILITIES.
        CLR     (R1)   ;END WITH CLEAN WORLD.
        MOV     @BIT14,(R1) ;MASTER CLEAR THE KMC.
        BIC    @BIT14!BIT15,(R1)

```

```

*****
*
*   WAIT CLOCK :- WAITS FOR KMC TO RESPOND
*   BACK WITH DONE OR ERROR IN SPECIFIC TIME IF DOES NOT
*   GET ONE GENERATES A TIME OUT ERROR.
*
*****

```

15290	036066	005037	036150	
15291	036072	005037	036146	

```

WATCLK: CLR      CLKCTR ;INITIALIZE CLOCK.
        CLR      DONE   ;WAIT.

```

```

1 036076 012711 100000      MOV      #BIT15,(R1)      ;INITIATE THE TEST...
1 036102 005737 036146      15:      TST      DONE          ;IS DONE FLAG SET?
1 036108 001010          BNE     35             ;YES, START THE NEXT TEST.
1 036110 005237 036150      INC     CLKCTR        ;COUNT BY ONE.
1 036114 005337 036150      DEC     CLKCTR        ;WAIT CLOCK.
1 036120 005237 036150      INC     CLKCTR
1 036126 001366          BNE     15             ;GO CHECK AGAIN. IF NOT TIMED OUT.
1 036130 000207          RTS     PC
1 036130 013737 001442 001206 35:      MOV     NEXT_SLPADR   ;SET THE NEXT TEST ADDRESS.
1 036130 012706 001200      MOV     @STACK,SP    ;POP THE STACK.
1 036134 000177 143040      JMP     @SLPADR      ;START THE NEXT TEST.
1 036134 000000          DONE:  0
1 036134 000000          CLKCTR:0
1 036134 000000          TSTRT:0
1 036134 000000          MCEND:0
1 036134 000000          MLDER: .ASCIZ <200>/LOADING ERROR/
1 036156 046200 040517 044504      .EVEN
1 036164 043516 042440 051122
1 036172 051117 000
1 036176          .FINI
1 036176 044600 052502 027523  EH1:  .ASCIZ <200>/IBUS/IBUS# REGISTER DATA TEST/
1 036235 200 041111 051525  EH2:  .ASCIZ <200>/IBUS/IBUS# REGISTER DUAL ADDRESSING TEST/
1 036237 200 051102 051040  EH3:  .ASCIZ <200>/M# REGISTER DATA TEST/
1 036237 051600 051103 052101  EH4:  .ASCIZ <200>/SCRATCH PAD DATA TEST/
1 036235 200 041523 040522  EH5:  .ASCIZ <200>/SCRATCH PAD DUAL ADDRESSING TEST/
1 036427 200 040515 047111  EH6:  .ASCIZ <200>/MAIN MEMORY DATA TEST/
1 036456 046600 044501 020116  EH7:  .ASCIZ <200>/MAIN MEMORY DUAL ADDRESSING TEST/
1 036520 040600 052125 020117  EH10: .ASCIZ <200>/AUTO INBINC FUNCTION TEST/
1 036523 200 050116 020122  EH11: .ASCIZ <200>/NPR TEST/
1 036523 200 052515 022114  EH12: .ASCIZ <200>/MULTIPLE NPR TEST/
1 036610 047200 047117 042440  EH13: .ASCIZ <200>/NON EX MEM FAILED TO SET/
1 036642 050200 047522 051107  EH14: .ASCIZ <200>/PROGRAM CLOCK TEST/
1 036666 040600 052514 043040  EH15: .ASCIZ <200>/ALU FUNCTION WITH C BIT CLEAR TEST/
1 036732 050200 053517 051106  EH16: .ASCIZ <200>/POWER FAIL; BUS INIT WAS NOT BLOCKED/
1 037000 043200 051117 042503  EH17: .ASCIZ <200>/FORCE POWER FAIL ERROR/
1 037030 047200 044517 042523  EH20: .ASCIZ <200>/NOISE TEST ON IBUS,IBUS,SPAD,MEMORY/
1 037076 040600 052514 041440  EH21: .ASCIZ <200>/ALU C BIT TEST FAILURE/
1 037126 052200 046511 020105  EH22: .ASCIZ <200>/TIME OUT ERROR/
1 037146 040600 052514 043040  EH23: .ASCIZ <200>/ALU FUNCTION TEST WITH C BIT SET/
1 037210 052600 041520 051440  EH24: .ASCIZ <200>/UPC SEQUENCE ERROR/
1 037234 044200 046105 020114  EH25: .ASCIZ <200>/HELL RAISER TEST/
1 037256 043600 047517 020104  DH1:  .ASCIZ <200>/GOOD      BAD      UPC      REGISTER/
1 037317 200 047507 042117  DH2:  .ASCIZ <200>/GOOD      BAD/
1 037334 043600 047517 020104  DH3:  .ASCIZ <200>/GOOD      BAD      UPC      ADDRESS/
1 037373 200 047507 042117  DH4:  .ASCIZ <200>/GOOD      BAD      UPC      FUNCTION/
1 037434          .EVEN
1 037434 000004      DT1:  4
1 037436 003 006      .BYTE  3,6
1 037440 001224          $GOODAT
1 037442 003 004      .BYTE  3,4
1 037444 001226          $BADAT
1 037446 003 006      .BYTE  3,6
1 037450 001222          $BADR
1 037452 003 006      .BYTE  3,6

```

F08

OZKCA MACY11 27(1006) 13-MAY-77 14:07 PAGE 281
OZKCA.P11 13-MAY-77 13:58 KMC11 ALU TESTS

PAGE: 0303

037454	001220		SGDADR	
037456	000002		DT2: 2	
037460	003	004	.BYTE	3,4
037462	001224		SGDDAT	
037464	003	004	.BYTE	3,4
037466	001226		SEDDAT	
037470	000001		CORMAX: .END	

ABASE =	000000	298	339		
ACOM1 =	000000	298	341		
ACOM2 =	000000	298	342		
ACPUOP =	000000	298	343		
ROOM0 =	000000	298	344		
ROOM1 =	000000	298	345		
ROOM10 =	000000	298	346		
ROOM11 =	000000	298	347		
ROOM12 =	000000	298	348		
ROOM13 =	000000	298	349		
ROOM14 =	000000	298	350		
ROOM15 =	000000	298	351		
ROOM2 =	000000	298	352		
ROOM3 =	000000	298	353		
ROOM4 =	000000	298	354		
ROOM5 =	000000	298	355		
ROOM6 =	000000	298	356		
ROOM7 =	000000	298	357		
ROOM8 =	000000	298	358		
ROOM9 =	000000	298	359		
ADEVCT =	000000	298	360		
ADEVN =	000000	298	361		
AORCHT =	006047	1400*	1415*	1424*	
ADVANC =	104420	1569*			
REIN =	000002	298			
REVN =	000000	298			
AFATAL =	000000	298			
ANPOL =	000000	298			
ANPOL2 =	000000	298			
ANPOL3 =	000000	298			
ANPOL4 =	000000	298			
ANPOL5 =	000000	298			
ANPOL6 =	000000	298			
ANPOL7 =	000000	298			
ANPOL8 =	000000	298			
ANPOL9 =	000000	298			
ANPOL10 =	000000	298			
ANPOL11 =	000000	298			
ANPOL12 =	000000	298			
ANPOL13 =	000000	298			
ANPOL14 =	000000	298			
ANPOL15 =	000000	298			
ANPOL16 =	000000	298			
ANPOL17 =	000000	298			
ANPOL18 =	000000	298			
ANPOL19 =	000000	298			
ANPOL20 =	000000	298			
ANPOL21 =	000000	298			
ANPOL22 =	000000	298			
ANPOL23 =	000000	298			
ANPOL24 =	000000	298			
ANPOL25 =	000000	298			
ANPOL26 =	000000	298			
ANPOL27 =	000000	298			
ANPOL28 =	000000	298			
ANPOL29 =	000000	298			
ANPOL30 =	000000	298			
ANPOL31 =	000000	298			
ANPOL32 =	000000	298			
ANPOL33 =	000000	298			
ANPOL34 =	000000	298			
ANPOL35 =	000000	298			
ANPOL36 =	000000	298			
ANPOL37 =	000000	298			
ANPOL38 =	000000	298			
ANPOL39 =	000000	298			
ANPOL40 =	000000	298			
ANPOL41 =	000000	298			
ANPOL42 =	000000	298			
ANPOL43 =	000000	298			
ANPOL44 =	000000	298			
ANPOL45 =	000000	298			
ANPOL46 =	000000	298			
ANPOL47 =	000000	298			
ANPOL48 =	000000	298			
ANPOL49 =	000000	298			
ANPOL50 =	000000	298			
ANPOL51 =	000000	298			
ANPOL52 =	000000	298			
ANPOL53 =	000000	298			
ANPOL54 =	000000	298			
ANPOL55 =	000000	298			
ANPOL56 =	000000	298			
ANPOL57 =	000000	298			
ANPOL58 =	000000	298			
ANPOL59 =	000000	298			
ANPOL60 =	000000	298			
ANPOL61 =	000000	298			
ANPOL62 =	000000	298			
ANPOL63 =	000000	298			
ANPOL64 =	000000	298			
ANPOL65 =	000000	298			
ANPOL66 =	000000	298			
ANPOL67 =	000000	298			
ANPOL68 =	000000	298			
ANPOL69 =	000000	298			
ANPOL70 =	000000	298			
ANPOL71 =	000000	298			
ANPOL72 =	000000	298			
ANPOL73 =	000000	298			
ANPOL74 =	000000	298			
ANPOL75 =	000000	298			
ANPOL76 =	000000	298			
ANPOL77 =	000000	298			
ANPOL78 =	000000	298			
ANPOL79 =	000000	298			
ANPOL80 =	000000	298			
ANPOL81 =	000000	298			
ANPOL82 =	000000	298			
ANPOL83 =	000000	298			
ANPOL84 =	000000	298			
ANPOL85 =	000000	298			
ANPOL86 =	000000	298			
ANPOL87 =	000000	298			
ANPOL88 =	000000	298			
ANPOL89 =	000000	298			
ANPOL90 =	000000	298			
ANPOL91 =	000000	298			
ANPOL92 =	000000	298			
ANPOL93 =	000000	298			
ANPOL94 =	000000	298			
ANPOL95 =	000000	298			
ANPOL96 =	000000	298			
ANPOL97 =	000000	298			
ANPOL98 =	000000	298			
ANPOL99 =	000000	298			
ANPOL100 =	000000	298			
APASS =	000000	298			
APRIOR =	000000	298			
APTCSU =	000040	1125	1230*		
APTENV =	000001	1118	1186	1228*	1630
APTSIZ =	000200	1227*			
APTSPO =	000100	1120	1188	1229*	
APT.SI =	013552	796	2210*		
ASMREC =	000000	298	311		
ATESTN =	000000	298	302		
ALDONE =	003354	833	854	893*	
ALUNIT =	000000	298	305		
ALSTRT =	003126	832*			
ALSMR =	000000	298	312		
AUTO.S =	012152	794	1954*		

DM3	037334	433	436	439	15312#									
DM4	037373	454	472	15312#										
DISPLA	001242	268#	758#	764#	1076#									
DISPRE	000174	229#	764											
DONE	036146	15168#	15182#	15291#	15293	15303#								
DSMR =	177570	82#	238#	267										
DT1	037434	419	422	428	431	434	437	440	455	473	15312#			
DT2	037456	425	443	446	464	479	15312#							
DZDME =	##### U	414												
DZDMG =	##### U	414												
DZKCA =	000000	1078	1089	1903	2275	2472	2669	2866	3064	3281	3514	3711	3908	4105
		4302	4499	4696	4893	5091	5268	5475	5682	5889	6096	6303	6510	6717
		6924	7131	7337	7511	7754	7891	8131	8288	8449	8662	8828	8954	9271
		9588	9905	10222	10539	10856	11173	11490	11807	12124	12441	12758	13075	13392
		13707	13850	14474	14692									
ENTVEC=	000030	171#												
EM1	036176	417	15312#											
EM10	036520	438	15312#											
EM11	036553	441	15312#											
EM12	036565	444	15312#											
EM13	036610	447	15312#											
EM14	036642	450	15312#											
EM15	036666	453	15312#											
EM16	036732	456	15312#											
EM17	037000	459	15312#											
EM2	036235	420	15312#											
EM20	037030	462	15312#											
EM21	037076	465	15312#											
EM22	037126	468	15312#											
EM23	037146	471	15312#											
EM24	037210	474	15312#											
EM25	037234	477	15312#											
EM3	036307	423	15312#											
EM4	036336	426	15312#											
EM5	036365	429	15312#											
EM6	036427	432	15312#											
EM7	036456	435	15312#											
ERCT00	002304	623#												
ERCT01	002310	626#												
ERCT02	002314	629#												
ERCT03	002320	632#												
ERCT04	002324	635#												
ERCT05	002330	638#												
ERCT06	002334	641#												
ERCT07	002340	644#												
ERCT10	002344	647#												
ERCT11	002350	650#												
ERCT12	002354	653#												
ERCT13	002360	656#												
ERCT14	002364	659#												
ERCT15	002370	662#												
ERCT16	002374	665#												
ERCT17	002400	668#												
EROR =	035350	2281#	2335	2415	2478#	2532	2612	2675#	2729	2809	2872#	2926	3006	3070#
		3133	3222	3287#	3359	3457	3520#	3574	3654	3717#	3771	3851	3914#	3968
		4048	4111#	4165	4245	4308#	4362	4442	4505#	4559	4639	4702#	4756	4836

CROSS REFERENCE TABLE -- USER SYMBOLS

KMS300 KMS301 KMS302 KMS303 KMS304 KMS305 KMS306 KMS307 KMS308 KMS309 KMS310 KMS311 KMS312 KMS313 KMS314 KMS315 KMS316 KMS317 KMS318 KMS319 KMS320 KMS321 KMS322 KMS323 KMS324 KMS325 KMS326 KMS327 KMS328 KMS329 KMS330 KMS331 KMS332 KMS333 KMS334 KMS335 KMS336 KMS337 KMS338 KMS339 KMS340 KMS341 KMS342 KMS343 KMS344 KMS345 KMS346 KMS347 KMS348 KMS349 KMS350 KMS351 KMS352 KMS353 KMS354 KMS355 KMS356 KMS357 KMS358 KMS359 KMS360 KMS361 KMS362 KMS363 KMS364 KMS365 KMS366 KMS367 KMS368 KMS369 KMS370 KMS371 KMS372 KMS373 KMS374 KMS375 KMS376 KMS377 KMS378 KMS379 KMS380 KMS381 KMS382 KMS383 KMS384 KMS385 KMS386 KMS387 KMS388 KMS389 KMS390 KMS391 KMS392 KMS393 KMS394 KMS395 KMS396 KMS397 KMS398 KMS399 KMS400 KMS401 KMS402 KMS403 KMS404 KMS405 KMS406 KMS407 KMS408 KMS409 KMS410 KMS411 KMS412 KMS413 KMS414 KMS415 KMS416 KMS417 KMS418 KMS419 KMS420 KMS421 KMS422 KMS423 KMS424 KMS425 KMS426 KMS427 KMS428 KMS429 KMS430 KMS431 KMS432 KMS433 KMS434 KMS435 KMS436 KMS437 KMS438 KMS439 KMS440 KMS441 KMS442 KMS443 KMS444 KMS445 KMS446 KMS447 KMS448 KMS449 KMS450 KMS451 KMS452 KMS453 KMS454 KMS455 KMS456 KMS457 KMS458 KMS459 KMS460 KMS461 KMS462 KMS463 KMS464 KMS465 KMS466 KMS467 KMS468 KMS469 KMS470 KMS471 KMS472 KMS473 KMS474 KMS475 KMS476 KMS477 KMS478 KMS479 KMS480 KMS481 KMS482 KMS483 KMS484 KMS485 KMS486 KMS487 KMS488 KMS489 KMS490 KMS491 KMS492 KMS493 KMS494 KMS495 KMS496 KMS497 KMS498 KMS499 KMS500 KMS501 KMS502 KMS503 KMS504 KMS505 KMS506 KMS507 KMS508 KMS509 KMS510 KMS511 KMS512 KMS513 KMS514 KMS515 KMS516 KMS517 KMS518 KMS519 KMS520 KMS521 KMS522 KMS523 KMS524 KMS525 KMS526 KMS527 KMS528 KMS529 KMS530 KMS531 KMS532 KMS533 KMS534 KMS535 KMS536 KMS537 KMS538 KMS539 KMS540 KMS541 KMS542 KMS543 KMS544 KMS545 KMS546 KMS547 KMS548 KMS549 KMS550 KMS551 KMS552 KMS553 KMS554 KMS555 KMS556 KMS557 KMS558 KMS559 KMS560 KMS561 KMS562 KMS563 KMS564 KMS565 KMS566 KMS567 KMS568 KMS569 KMS570 KMS571 KMS572 KMS573 KMS574 KMS575 KMS576 KMS577 KMS578 KMS579 KMS580 KMS581 KMS582 KMS583 KMS584 KMS585 KMS586 KMS587 KMS588 KMS589 KMS590 KMS591 KMS592 KMS593 KMS594 KMS595 KMS596 KMS597 KMS598 KMS599 KMS600 KMS601 KMS602 KMS603 KMS604 KMS605 KMS606 KMS607 KMS608 KMS609 KMS610 KMS611 KMS612 KMS613 KMS614 KMS615 KMS616 KMS617 KMS618 KMS619 KMS620 KMS621 KMS622 KMS623 KMS624 KMS625 KMS626 KMS627 KMS628 KMS629 KMS630 KMS631 KMS632 KMS633 KMS634 KMS635 KMS636 KMS637 KMS638 KMS639 KMS640 KMS641 KMS642 KMS643 KMS644 KMS645 KMS646 KMS647 KMS648 KMS649 KMS650 KMS651 KMS652 KMS653 KMS654 KMS655 KMS656 KMS657 KMS658 KMS659 KMS660 KMS661 KMS662 KMS663 KMS664 KMS665 KMS666 KMS667 KMS668 KMS669 KMS670 KMS671 KMS672 KMS673 KMS674 KMS675 KMS676 KMS677 KMS678 KMS679 KMS680 KMS681 KMS682 KMS683 KMS684 KMS685 KMS686 KMS687 KMS688 KMS689 KMS690 KMS691 KMS692 KMS693 KMS694 KMS695 KMS696 KMS697 KMS698 KMS699 KMS700 KMS701 KMS702 KMS703 KMS704 KMS705 KMS706 KMS707 KMS708 KMS709 KMS710 KMS711 KMS712 KMS713 KMS714 KMS715 KMS716 KMS717 KMS718 KMS719 KMS720 KMS721 KMS722 KMS723 KMS724 KMS725 KMS726 KMS727 KMS728 KMS729 KMS730 KMS731 KMS732 KMS733 KMS734 KMS735 KMS736 KMS737 KMS738 KMS739 KMS740 KMS741 KMS742 KMS743 KMS744 KMS745 KMS746 KMS747 KMS748 KMS749 KMS750 KMS751 KMS752 KMS753 KMS754 KMS755 KMS756 KMS757 KMS758 KMS759 KMS760 KMS761 KMS762 KMS763 KMS764 KMS765 KMS766 KMS767 KMS768 KMS769 KMS770 KMS771 KMS772 KMS773 KMS774 KMS775 KMS776 KMS777 KMS778 KMS779 KMS780 KMS781 KMS782 KMS783 KMS784 KMS785 KMS786 KMS787 KMS788 KMS789 KMS790 KMS791 KMS792 KMS793 KMS794 KMS795 KMS796 KMS797 KMS798 KMS799 KMS800 KMS801 KMS802 KMS803 KMS804 KMS805 KMS806 KMS807 KMS808 KMS809 KMS810 KMS811 KMS812 KMS813 KMS814 KMS815 KMS816 KMS817 KMS818 KMS819 KMS820 KMS821 KMS822 KMS823 KMS824 KMS825 KMS826 KMS827 KMS828 KMS829 KMS830 KMS831 KMS832 KMS833 KMS834 KMS835 KMS836 KMS837 KMS838 KMS839 KMS840 KMS841 KMS842 KMS843 KMS844 KMS845 KMS846 KMS847 KMS848 KMS849 KMS850 KMS851 KMS852 KMS853 KMS854 KMS855 KMS856 KMS857 KMS858 KMS859 KMS860 KMS861 KMS862 KMS863 KMS864 KMS865 KMS866 KMS867 KMS868 KMS869 KMS870 KMS871 KMS872 KMS873 KMS874 KMS875 KMS876 KMS877 KMS878 KMS879 KMS880 KMS881 KMS882 KMS883 KMS884 KMS885 KMS886 KMS887 KMS888 KMS889 KMS890 KMS891 KMS892 KMS893 KMS894 KMS895 KMS896 KMS897 KMS898 KMS899 KMS900 KMS901 KMS902 KMS903 KMS904 KMS905 KMS906 KMS907 KMS908 KMS909 KMS910 KMS911 KMS912 KMS913 KMS914 KMS915 KMS916 KMS917 KMS918 KMS919 KMS920 KMS921 KMS922 KMS923 KMS924 KMS925 KMS926 KMS927 KMS928 KMS929 KMS930 KMS931 KMS932 KMS933 KMS934 KMS935 KMS936 KMS937 KMS938 KMS939 KMS940 KMS941 KMS942 KMS943 KMS944 KMS945 KMS946 KMS947 KMS948 KMS949 KMS950 KMS951 KMS952 KMS953 KMS954 KMS955 KMS956 KMS957 KMS958 KMS959 KMS960 KMS961 KMS962 KMS963 KMS964 KMS965 KMS966 KMS967 KMS968 KMS969 KMS970 KMS971 KMS972 KMS973 KMS974 KMS975 KMS976 KMS977 KMS978 KMS979 KMS980 KMS981 KMS982 KMS983 KMS984 KMS985 KMS986 KMS987 KMS988 KMS989 KMS990 KMS991 KMS992 KMS993 KMS994 KMS995 KMS996 KMS997 KMS998 KMS999 KMS1000	002108 002109 002110 002111 002112 002113 002114 002115 002116 002117 002118 002119 002120 002121 002122 002123 002124 002125 002126 002127 002128 002129 002130 002131 002132 002133 002134 002135 002136 002137 002138 002139 002140 002141 002142 002143 002144 002145 002146 002147 002148 002149 002150 002151 002152 002153 002154 002155 002156 002157 002158 002159 002160 002161 002162 002163 002164 002165 002166 002167 002168 002169 002170 002171 002172 002173 002174 002175 002176 002177 002178 002179 002180 002181 002182 002183 002184 002185 002186 002187 002188 002189 002190 002191 002192 002193 002194 002195 002196 002197 002198 002199 002200 002201 002202 002203 002204 002205 002206 002207 002208 002209 002210 002211 002212 002213 002214 002215 002216 002217 002218 002219 002220 002221 002222 002223 002224 002225 002226 002227 002228 002229 002230 002231 002232 002233 002234 002235 002236 002237 002238 002239 002240 002241 002242 002243 002244 002245 002246 002247 002248 002249 002250 002251 002252 002253 002254 002255 002256 002257 002258 002259 002260 002261 002262 002263 002264 002265 002266 002267 002268 002269 002270 002271 002272 002273 002274 002275 002276 002277 002278 002279 002280 002281 002282 002283 002284 002285 002286 002287 002288 002289 002290 002291 002292 002293 002294 002295 002296 002297 002298 002299 002300 002301 002302 002303 002304 002305 002306 002307 002308 002309 002310 002311 002312 002313 002314 002315 002316 002317 002318 002319 002320 002321 002322 002323 002324 002325 002326 002327 002328 002329 002330 002331 002332 002333 002334 002335 002336 002337 002338 002339 002340 002341 002342 002343 002344 002345 002346 002347 002348 002349 002350 002351 002352 002353 002354 002355 002356 002357 002358 002359 002360 002361 002362 002363 002364 002365 002366 002367 002368 002369 002370 002371 002372 002373 002374 002375 002376 002377 002378 002379 002380 002381 002382 002383 002384 002385 002386 002387 002388 002389 002390 002391 002392 002393 002394 002395 002396 002397 002398 002399 002400 002401 002402 002403 002404 002405 002406 002407 002408 002409 002410 002411 002412 002413 002414 002415 002416 002417 002418 002419 002420 002421 002422 002423 002424 002425 002426 002427 002428 002429 002430 002431 002432 002433 002434 002435 002436 002437 002438 002439 002440 002441 002442 002443 002444 002445 002446 002447 002448 002449 002450 002451 002452 002453 002454 002455 002456 002457 002458 002459 002460 002461 002462 002463 002464 002465 002466 002467 002468 002469 002470 002471 002472 002473 002474 002475 002476 002477 002478 002479 002480 002481 002482 002483 002484 002485 002486 002487 002488 002489 002490 002491 002492 002493 002494 002495 002496 002497 002498 002499 002500 002501 002502 002503 002504 002505 002506 002507 002508 002509 002510 002511 002512 002513 002514 002515 002516 002517 002518 002519 002520 002521 002522 002523 002524 002525 002526 002527 002528 002529 002530 002531 002532 002533 002534 002535 002536 002537 002538 002539 002540 002541 002542 002543 002544 002545 002546 002547 002548 002549 002550 002551 002552 002553 002554 002555 002556 002557 002558 002559 002560 002561 002562 002563 002564 002565 002566 002567 002568 002569 002570 002571 002572 002573 002574 002575 002576 002577 002578 002579 002580 002581 002582 002583 002584 002585 002586 002587 002588 002589 002590 002591 002592 002593 002594 002595 002596 002597 002598 002599 002600 002601 002602 002603 002604 002605 002606 002607 002608 002609 002610 002611 002612 002613 002614 002615 002616 002617 002618 002619 002620 002621 002622 002623 002624 002625 002626 002627 002628 002629 002630 002631 002632 002633 002634 002635 002636 002637 002
---	---

CROSS REFERENCE TABLE -- USER SYMBOLS

OTNPR = 000020	1#								
OUISR = 035350	1905#	15162#							
PACT00 = 002302	622#								
PACT01 = 002305	625#								
PACT02 = 002312	628#								
PACT03 = 002316	631#								
PACT04 = 002322	634#								
PACT05 = 002325	637#								
PACT06 = 002332	640#								
PACT07 = 002336	643#								
PACT10 = 002342	646#								
PACT11 = 002346	649#								
PACT12 = 002350	652#								
PACT13 = 002356	655#								
PACT14 = 002362	658#								
PACT15 = 002366	661#								
PACT16 = 002372	664#								
PACT17 = 002376	667#								
PAGED = 000000	1#	14456	14876	14946	15001				
PAGE1 = 001000	1#								
PAGE2 = 002000	1#								
PAGE3 = 003000	1#								
PARBIT = 040000	191#								
PCLX = 000020	1#								
PERFOR = 004537	191#								
PFTAB = 007276	1688#	1710#							
PIAQ = 177772	81#								
PIARVE = 000240	175#								
POPAD = 012600	186#	1643							
POP1SP = 005726	184#								
POP2SP = 022626	188#	15129							
PRIO = 010374	1769#	1996							
PRIRTY = 013772	2217#	2223#	2236	2257#					
PRO = 000000	98#	2193	2194						
PR1 = 000040	99#								
PR2 = 000100	100#								
PR3 = 000140	101#								
PR4 = 000200	102#	1907	2195	15093					
PR5 = 000240	103#	1904	1906	2196					
PR6 = 000300	104#	2197							
PR7 = 000340	105#	210	212	214	216	2198			
PS = 177776	78#	79	732*	941*	1907*	2162*	2172*	15093*	
PSM = 177776	79#								
PUSHAD = 010046	185#	1640							
PUSHLS = 005746	183#								
PUSHES = 024646	187#								
PWAIL = 025134	15070	15111#							
PWVEC = 000024	170#	1666#	1667*	1676*	1682*	1701*	1702*		
QV.FLG = 001511	394#	738#	988*						
RABO = 000200	1#								
RACT = 000100	1#								
RACC = 000001	1#								
RBR = 000200	1#								
ROCHR = 104402	1282	1555#							
ROLIN = 104403	1355	1556#							
ROOCT = 104404	1404	1557#							

REOM = 000002	1629	1629															
RESVEC = 000010	1650																
RESOS = 104407	1560	1629															
RETN1 = 035330	15074	15098	15106	15145													
RETN2 = 035316	15139	15140	15141	15143													
RET1 = 035506	15169	15174	15183														
RET2 = 035516	15178	15185															
RONCLK = 104412	1563	1716	1719	1756	1761												
RONPR = 000001	1																
RQDY = 000020	1																
RUN = 001900	394	741	1867	1868	1875												
SAVACT = 001474	382	901	2133	2225													
SAVNUN = 001476	383	735	986	989	2126	2219											
SAVPC = 001460	376	862	877	1436	1655	15086											
SAVSP = 001456	375																
SAVOS = 104406	1559	1588															
SCOP1 = 104405	1558	15100															
SCPD1 = 035106	15068	15099															
SCPD11 = 035126	15102	15104															
SCPE = 035500	2281	2451	2478	2648	2675	2845	2872	3042	3070	3258	3287	3493	3520				
	3690	3717	3887	3914	4084	4111	4281	4308	4478	4505	4675	4702	4872				
	4899	5069	5097	5274	5274	5454	5481	5661	5688	5868	5895	6075	6102				
	6282	6309	6489	6516	6696	6723	6903	6930	7110	7137	7317	7343	7489				
	7517	7732	7760	7897	7897	8137	8137	8294	8294	8455	8455	8641	8668				
	8807	8834	8930	8960	9247	9277	9564	9594	9881	9911	10198	10228	10515				
	10545	10832	10862	11149	11179	11466	11496	11783	11813	12100	12130	12417	12447				
	12734	12764	13051	13081	13368	13398	13685	13713	13826	13856	14412	14480	14668				
	14698																
SCPE0 = 034570	14954	15045															
SCPE1 = 001130	2	2281	2478	2675	2872	3070	3287	3520	3717	3914	4111	4308	4505				
	4702	4899	5097	5274	5481	5688	5895	6102	6309	6516	6723	6930	7137				
	7343	7517	7760	7897	8137	8294	8455	8668	8834	8960	9277	9594	9911				
	10228	10545	10862	11179	11496	11813	12130	12447	12764	13081	13398	13713	13856				
	14480	14698															
SCP1 = 035426	2281	2478	2478	2675	2675	2742	2822	2872	2939	2939	3019	3070					
	3146	3235	3287	3372	3470	3520	3587	3667	3717	3784	3864	3981					
	4061	4111	4178	4258	4308	4375	4455	4505	4572	4652	4702	4849					
	4899	4966	5046	5097	5274	5346	5431	5481	5553	5638	5688	5760	5845				
	5895	5967	6052	6102	6174	6259	6309	6381	6466	6516	6588	6673	6723				
	6795	6880	6930	7002	7087	7137	7209	7294	7343	7517	7588	7673	7723				
	8218	8294	8375	8455	8668	8772	8834	8960	9149	9220	9277	9466	9537				
	9594	9783	9854	9911	10100	10171	10228	10417	10488	10545	10734	10805	10862				
	11051	11122	11179	11368	11439	11496	11685	11756	11813	12002	12073	12130	12319				
	12390	12447	12636	12707	12764	12953	13024	13081	13270	13341	13398	13587	13658				
	13713	13856	14480	14698													
SCP10 = 034516	14884																
SCP11 = 035430	2281	2478	2675	2872	3070	3287	3520	3717	3914	4111	4308	4505	4702				
	4899	5097	5151	5221	5274	5481	5688	5895	6102	6309	6516	6723	6930				
	7137	7343	7517	7760	7897	8137	8294	8455	8525	8609	8668	8834	8896				
	8960	9277	9594	9911	10228	10545	10862	11179	11496	11813	12130	12447	12764				
	13081	13398	13713	13856	14480	14698											
SCP110 = 034520	14887																
SCP111 = 001056	2	2281	2478	2675	2872	3070	3287	3520	3717	3914	4111	4308	4505				
	4702	4899	5097	5274	5481	5688	5895	6102	6309	6516	6723	6930	7137				
	7343	7517	7760	7897	8137	8294	8455	8668	8834	8960	9277	9594	9911				

TPVEC = 000064	174#			
TRAPVE = 000034	172#			
TRTVEC = 000014	167#			
TSTEN 034440	14789#	15223	15260	
TSTRT 036153	15218#	15256	15305#	
TST1 013774	1040	1923	1941	2272#
TST10 015674	3512	3708#		
TST11 016100	3709	3905#		
TST12 016304	3906	4102#		
TST13 016510	4103	4299#		
TST14 016714	4300	4496#		
TST15 017120	4497	4693#		
TST16 017324	4694	4890#		
TST17 017530	4891	5088#		
TST2 014200	2273	2469#		
TST20 017720	5089	5265#		
TST21 020134	5266	5472#		
TST22 020350	5473	5679#		
TST23 020564	5680	5886#		
TST24 021000	5887	6093#		
TST25 021214	6094	6300#		
TST26 021430	6301	6507#		
TST27 021644	6508	6714#		
TST3 014404	2470	2666#		
TST30 022060	6715	6921#		
TST31 022274	6922	7128#		
TST32 022510	7129	7334#		
TST33 022704	7335	7508#		
TST34 023154	7509	7751#		
TST35 023322	7752	7989#		
TST36 023606	7989	8129#		
TST37 023764	8129	8285#		
TST4 014610	2667	2863#		
TST40 024144	8286	8446#		
TST41 024362	8447	8659#		
TST42 024544	8660	8825#		
TST43 024700	8826	8951#		
TST44 025226	8952	9268#		
TST45 025554	9269	9585#		
TST46 026102	9586	9902#		
TST47 026430	9903	10219#		
TST5 015014	2864	3061#		
TST50 026756	10220	10536#		
TST51 027304	10537	10853#		
TST52 027632	10854	11170#		
TST53 030160	11171	11487#		
TST54 030506	11488	11804#		
TST55 031034	11805	12121#		
TST56 031362	12122	12438#		
TST57 031710	12439	12755#		
TST6 015234	3062	3278#		
TST60 032236	12756	13072#		
TST61 032564	13073	13389#		
TST62 033112	13390	13704#		
TST63 033260	13705	13947#		
TST64 034064	13948	14471#		

CROSS REFERENCE TABLE -- USER SYMBOLS

		8132	8136	8137#	8274	8280	8283	8288	8289	8293	8294#	8434	8441	8444
		8149	8150	8154	8155#	8247	8254	8257	8262	8263	8267	8268#	8270	8270
		8223	8228	8229	8233	8234#	8236	8246	8249	8254	8255	8259	8260	8263
		8263	8266	8271	8272	8276	8277#	8270	8280	8283	8288	8294#	8434	8441
		8287	8297	8300	8305	8306	8310	8311#	8320	8323	8328	8329	8333	8334
		10228#	10521	10531	10534	10539	10540	10544	10545#	10838	10848	10851	10856	10857
		10861	10862#	11155	11165	11169	11173	11174	11178	11179#	11472	11482	11485	11490
		11491	11495#	11496#	11789	11799	11802	11807	11808	11812	11813#	12106	12116	12119
		12124	12125	12129	12130#	12423	12433	12436	12441	12442	12446	12447#	12740	12750
		12753	12758	12759	12763	12764#	13057	13067	13070	13075	13076	13080	13081#	13374
		13384	13387	13392	13393	13397	13398#	13691	13699	13702	13707	13708	13712	13713#
		13832	13842	13845	13850	13851	13855	13856#	14459	14466	14469	14474	14475	14479
		14480#	14675	14683	14687	14692	14693	14697	14698#	15312#				
\$NULL	001254	273#	1145	1174										
\$OVER	004334	1044	1047	1058	1070	1076#								
\$PASS	001324	303#	968#	980	992#	993#	1010	1018	1066	1082	1885*			
\$PASTH	002042	500#												
\$PARDN	007100	211	734	1666#	1701	15133	15136							
\$PLUNG	007264	1704#												
\$PWLUP	007152	1676	1682#											
\$QUES	001312	291#	1174	1312	1329	1386	1389	1409	1938	2010	2024	2038		
\$ROCHR	005134	1249#	1555											
\$RODEC	##### U	1558												
\$ROLIN	005254	1277#	1556											
\$ROOCT	005554	1350#	1557											
\$ROSZ =	000007	1270#												
\$REGAD	001260	277#												
\$REGO	001262	279#	1445*	1450										
\$REG1	001264	280#	868#	880	1444*	1451								
\$REG2	001266	281#	1443#	1452										
\$REG3	001270	282#	1442#	1453										
\$REG4	001272	283#	1441#	1454										
\$REG5	001274	284#	1440#	1455										
\$RTNAD	004102	1009#												
\$R2A =	##### U	1558												
\$S	= 000067	1#	2273	2281#	2470	2478#	2667	2675#	2864	2872#	3062	3070#	3279	3287#
		3512	3520#	3709	3717#	3906	3914#	4103	4111#	4300	4308#	4497	4505#	4694
		4702#	4891	4899#	5089	5097#	5266	5274#	5473	5481#	5680	5688#	5887	5895#
		6094	6102#	6301	6309#	6508	6516#	6715	6723#	6922	6930#	7129	7137#	7335
		7343#	7509	7517#	7752	7760#	7889	7897#	8129	8137#	8286	8294#	8447	8455#
		8660	8668#	8826	8834#	8952	8960#	9269	9277#	9586	9594#	9903	9911#	10220
		10228#	10537	10545#	10854	10862#	11171	11179#	11488	11496#	11805	11813#	12122	12130#
		12439	12447#	12756	12764#	13073	13081#	13390	13398#	13705	13713#	13848	13856#	14472
		14480#	14690	14698#										
\$SAVRE =	##### U	1558												
\$SAVR6	007274	1675#	1683	1684#	1685#	1708#								
\$SCOPE	004134	209	1038#	2190										
\$SETUP =	000000	991	1039	1238	1335									
\$SVLAD	004316	1055	1073#											
\$SVPC =	000040	221#	226											
\$SMR =	164000	1#	42	290	291	963	991	1002	1008	1010	1032	1033	1034	1035
		1046	1058	1060	1061	1062	1063	1064	1076	1081	1705			
\$SMREG	001340	311#	752											
\$SMRANK =	000000	1035												
\$TESTN	001322	302#	1074*											
\$TRES	001210	290#	991*	1063*	1069	1072*	1081							

CROSS REFERENCE TABLE -- USER SYMBOLS

STK8	001246	270#	1045	1236	1253	1259	1798	1800	1842	2203				
STKS	001244	269#	1043	1236	1251	1257	1840	2201						
STNPO	001276	285#	807#	1771	2158#	2159#								
STNP1	001300	286#	808#	1773										
STNP2	001302	287#	810#	831#	858	867#	1775	1971	1974	2061#				
STNP3	001304	288#	811#	1777	1983	1986	1991	1994	2048	2051	2056	2059		
STNP4	001306	289#	812#	1779	1966#	1978#	2144							
STN	= 000001	1#	42											
STPB	001252	272#	1163#	1174	1581#	1845#	2206#							
STPFLG	001257	276#	1112	1174										
STPS	001250	271#	1161	1174	1579	1843	2204							
STRAP	006404	215	1529#											
STRAP2	006426	1540#	1551											
STRP	= 000021	1544#	1553#	1555	1556#	1557#	1558#	1559#	1560#	1561#	1562#	1563#	1564#	1565#
		1566#	1567#	1568#	1569#	1570#								
STRPAD	006440	1534	1551#											
STSTM	002040	499#												
STSTN1	001202	249#	747#	1031	1073#	1074	1076	1082	1658	1712	1920	1927	1929	2272#
		2469#	2666#	2863#	3061#	3278#	3511#	3708#	3905#	4102#	4299#	4496#	4693#	4890#
		5088#	5265#	5472#	5679#	5886#	6093#	6300#	6507#	6714#	6921#	7128#	7334#	7508#
		7751#	7888#	8128#	8285#	8446#	8659#	8825#	8951#	9268#	9585#	9902#	10219#	10536#
		10853#	11170#	11487#	11804#	12121#	12438#	12755#	13072#	13389#	13704#	13847#	14471#	14689#
STYIN	005510	1279	1280	1292	1310	1324	1328#							
STYBN	##### U	1553												
STYPS	##### U	1553												
STYPC	004404	1112#	1205	1544	1552									
STYPEC	004616	1142	1149	1156	1161#	1162								
STYPEX	004664	1167	1169	1172#										
STYPC	##### U	1553												
SUNIT	001330	305#												
SUNITH	002044	501#												
SUSR	001342	312#												
SVECT1	001366	337#	2222	2223										
SVECT2	001370	338#												
SXTSTR	004174	1041	1049#											
SY	= 000000	1#	503#											
SSGET4	= 000000	1002#												
SSINIT	= 033440	2281#	2310	2335	2341	2348	2354	2363	2366	2390	2415	2421	2428	2434
		2443	2451	2454	2478#	2507	2532	2538#	2545	2551	2560	2563	2587	2612
		2618	2625	2631	2640	2648	2651	2675#	26704	2729	2735	2743	2748	2757
		2760	2784	2809	2815	2823	2828	2837#	2845	2848	2872#	2901	2926	2932
		2939	2945	2954	2957	2981	2988	3006	3012	3019	3023	3034	3045	3070#
		3108	3133	3139	3146	3152	3161	3164	3164	3197	3223	3228	3241	3250
		3258	3261	3287#	3334	3359	3365	3372	3378	3387	3390	3432	3457	3463
		3470	3476	3485	3493	3496	3520#	3549	3574	3580	3587	3593	3602	3605
		3629	3654	3660	3667	3673	3682	3690	3693	3717#	3746	3771	3777	3784
		3790	3799	3802	3826	3851	3857	3864	3870	3879	3887	3890	3914#	3943
		3968	3974	3981	3987	3996	3999	4023	4048	4054	4061	4067	4076	4084
		4087	4111#	4140	4165	4171	4178	4184	4193	4196	4220	4245	4251	4258
		4264	4273	4281	4284	4308#	4337	4362	4368	4375	4381	4390	4393	4417
		4442	4448	4455	4461	4470	4478	4481	4505#	4534	4559	4565	4572	4578
		4587	4590	4614	4639	4645	4652	4658	4667	4675	4678	4702#	4731	4756
		4762	4769	4775	4784	4787	4811	4836	4842	4849	4855	4864	4872	4875
		4899#	4928	4953	4959	4966	4972	4981	4984	5008	5033	5039	5046	5052
		5061	5069	5072	5097#	5125	5141	5144	5151	5154	5169	5172	5195	5211
		5214	5221	5224	5239	5247	5250	5274#	5305	5333	5339	5346	5352	5361

CROSS REFERENCE TABLE -- USER SYMBOLS

53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	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	704	705	706	707	708	709	710	711	712	713	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	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	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	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	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	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	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	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503	1504	1505	1506	1507	1508	1509</
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	--------

CROSS REFERENCE TABLE -- USER SYMBOLS

3083	3086	3089	3092	3095	3098	3104	3110	3113	3116	3119	3122	3125
3129	3135	3142	3148	3154	3168	3172	3175	3178	3181	3184	3187	3193
3199	3202	3205	3208	3211	3214	3218	3224	3231	3237	3243	3254	3269
3293	3298	3300	3303	3306	3309	3312	3315	3318	3321	3324	3330	3336
3339	3342	3345	3348	3351	3355	3361	3368	3374	3380	3394	3398	3401
3404	3407	3410	3413	3416	3419	3422	3428	3434	3437	3440	3443	3446
3449	3453	3459	3465	3472	3478	3489	3492	3495	3499	3503	3507	3509
3545	3551	3554	3557	3560	3563	3567	3570	3576	3583	3589	3596	3609
3613	3616	3619	3625	3631	3634	3637	3640	3643	3646	3650	3656	3663
3669	3675	368	3687	3693	3696	3700	3703	3706	3709	3715	3751	3754
3757	3760	3763	3767	3773	3780	3786	3789	3795	3800	3810	3816	3822
3828	3831	3834	3837	3840	3843	3847	3853	3860	3865	3872	3883	3896
3919	3923	3927	3930	3933	3939	3945	3948	3951	3954	3957	3960	3964
3970	3977	3983	3989	3993	4007	4010	4013	4019	4025	4028	4031	4034
4037	4040	4044	4050	4057	4063	4069	4080	4098	4113	4120	4124	4127
4130	4136	4142	4145	4148	4151	4154	4157	4161	4167	4174	4180	4186
4200	4204	4207	4210	4216	4222	4225	4228	4231	4234	4237	4241	4247
4254	4260	4266	4277	4310	4313	4317	4321	4324	4327	4333	4339	4342
4345	4348	4351	4354	4358	4364	4371	4377	4383	4387	4393	4404	4407
4413	4419	4422	4425	4428	4431	4434	4438	4444	4451	4457	4463	4474
4507	4510	4514	4518	4521	4524	4530	4536	4539	4542	4545	4548	4551
4555	4561	4568	4574	4580	4584	4594	4601	4604	4610	4616	4619	4622
4625	4628	4631	4635	4641	4648	4654	4660	4671	4704	4707	4711	4715
4718	4721	4727	4733	4736	4739	4742	4745	4748	4752	4758	4765	4771
4777	4791	4795	4798	4801	4807	4813	4816	4819	4822	4825	4828	4832
4838	4845	4851	4857	4864	4901	4904	4908	4912	4915	4918	4924	4930
4933	4936	4939	4942	4945	4949	4955	4962	4968	4974	4980	4982	4985
4998	5004	5010	5013	5016	5019	5022	5025	5029	5035	5042	5048	5054
5065	5099	5102	5106	5109	5112	5115	5121	5127	5130	5133	5137	5147
5156	5165	5168	5179	5182	5185	5191	5197	5200	5203	5207	5217	5228
5235	5243	5247	5250	5253	5256	5262	5269	5275	5280	5287	5294	5305
5316	5319	5325	5327	5330	5333	5336	5342	5349	5350	5357	5374	5377
5380	5383	5386	5389	5392	5395	5398	5404	5410	5414	5417	5420	5423
5429	5435	5438	5441	5444	5447	5453	5459	5465	5471	5474	5477	5480
5487	5493	5496	5499	5505	5511	5514	5517	5523	5529	5535	5540	5547
5546	5557	5563	5569	5575	5581	5584	5587	5593	5599	5605	5610	5617
5646	5657	5663	5669	5675	5681	5684	5687	5693	5699	5705	5711	5717
5730	5733	5736	5739	5743	5749	5755	5762	5768	5774	5780	5786	5791
5794	5800	5806	5812	5815	5818	5821	5827	5833	5839	5844	5847	5854
5853	5864	5877	5902	5904	5907	5910	5913	5916	5922	5928	5931	5934
5937	5940	5943	5946	5950	5956	5963	5969	5975	5982	5988	5995	5998
6001	6007	6013	6016	6019	6022	6025	6028	6031	6035	6041	6048	6054
6060	6071	6104	6127	6144	6157	6163	6170	6182	6189	6199	6205	6205
6144	6147	6150	6153	6157	6163	6170	6182	6189	6196	6199	6205	6205
6208	6214	6220	6223	6226	6229	6232	6235	6238	6242	6248	6255	6261
6267	6278	6311	6314	6318	6321	6324	6327	6330	6336	6342	6348	6348
6351	6354	6357	6360	6364	6370	6377	6383	6389	6395	6403	6409	6412
6415	6421	6427	6430	6433	6436	6439	6442	6445	6449	6455	6463	6468
6474	6485	6518	6521	6525	6528	6531	6534	6537	6543	6549	6555	6555
6558	6561	6564	6567	6571	6577	6584	6590	6596	6603	6610	6616	6619
6622	6628	6634	6637	6640	6643	6646	6649	6652	6658	6665	6675	6675
6681	6682	6725	6728	6732	6735	6738	6741	6744	6750	6756	6762	6762
6765	6768	6771	6774	6778	6784	6791	6797	6803	6809	6817	6823	6826
6829	6835	6841	6844	6847	6850	6853	6856	6859	6863	6869	6875	6875
6888	6899	6932	6935	6939	6942	6945	6948	6951	6957	6963	6975	6975
6972	6975	6978	6981	6985	6991	6998	7004	7010	7024	7027	7030	7033

CROSS REFERENCE TABLE -- USER SYMBOLS

7036	7042	7049	7051	7054	7057	7060	7063	7066	7070	7076	7083	7089
7037	7106	7129	7148	7146	7149	7152	7155	7158	7164	7170	7173	7176
7179	7180	7186	7188	7192	7198	7205	7211	7217	7221	7224	7227	7240
7254	7249	7250	7257	7261	7264	7267	7270	7273	7277	7282	7290	7296
7300	7313	7314	7317	7320	7323	7326	7329	7332	7335	7338	7341	7377
7380	7390	7391	7392	7395	7398	7401	7404	7407	7410	7413	7419	7425
7428	7431	7434	7437	7440	7443	7446	7449	7452	7455	7471	7475	7485
7518	7519	7520	7523	7526	7529	7532	7535	7538	7541	7544	7551	7554
7557	7559	7560	7563	7566	7569	7572	7575	7578	7581	7584	7590	7596
7599	7602	7605	7608	7611	7614	7617	7620	7623	7626	7629	7636	7641
7647	7650	7653	7656	7659	7662	7665	7668	7671	7674	7677	7684	7702
7705	7708	7711	7714	7717	7720	7723	7726	7729	7732	7735	7742	7779
7782	7785	7788	7791	7794	7797	7800	7803	7806	7809	7812	7815	7818
7821	7824	7827	7830	7833	7836	7839	7842	7845	7848	7851	7854	7857
7910	7913	7916	7919	7922	7925	7928	7931	7934	7937	7940	7943	7946
7951	7954	7957	7960	7963	7966	7969	7972	7975	7978	7981	7984	7987
8021	8024	8027	8030	8033	8036	8039	8042	8045	8048	8051	8054	8057
8061	8064	8067	8070	8073	8076	8079	8082	8085	8088	8091	8094	8097
8148	8167	8170	8173	8176	8179	8182	8185	8188	8191	8194	8197	8214
8220	8235	8241	8247	8253	8259	8265	8271	8277	8283	8289	8295	8311
8314	8317	8321	8327	8333	8339	8345	8351	8357	8363	8369	8375	8391
8394	8371	8377	8383	8389	8395	8401	8407	8413	8419	8425	8431	8447
8450	8471	8477	8483	8489	8495	8501	8507	8513	8519	8525	8531	8547
8567	8591	8597	8603	8609	8615	8621	8627	8633	8639	8645	8651	8667
8748	8751	8757	8763	8769	8775	8781	8787	8793	8799	8805	8811	8817
8847	8850	8856	8862	8868	8874	8880	8886	8892	8898	8904	8910	8916
8926	8931	8937	8943	8949	8955	8961	8967	8973	8979	8985	8991	8997
9000	9003	9009	9015	9021	9027	9033	9039	9045	9051	9057	9063	9069
9078	9081	9087	9093	9099	9105	9111	9117	9123	9129	9135	9141	9147
9128	9132	9138	9144	9150	9156	9162	9168	9174	9180	9186	9192	9198
9190	9193	9199	9205	9211	9217	9223	9229	9235	9241	9247	9253	9259
9287	9290	9296	9302	9308	9314	9320	9326	9332	9338	9344	9350	9356
9368	9371	9377	9383	9389	9395	9401	9407	9413	9419	9425	9431	9437
9439	9415	9418	9424	9430	9436	9442	9448	9454	9460	9466	9472	9478
9468	9474	9477	9483	9489	9495	9501	9507	9513	9519	9525	9531	9537
9520	9533	9533	9539	9545	9551	9557	9563	9569	9575	9581	9587	9593
9619	9622	9628	9634	9640	9646	9652	9658	9664	9670	9676	9682	9688
9658	9651	9657	9663	9669	9675	9681	9687	9693	9699	9705	9711	9717
9697	9700	9703	9709	9715	9721	9727	9733	9739	9745	9751	9757	9763
9747	9750	9753	9759	9765	9771	9777	9783	9789	9795	9801	9807	9813
9803	9806	9812	9818	9824	9830	9836	9842	9848	9854	9860	9866	9872
9862	9877	9892	9907	9915	9920	9926	9932	9938	9944	9950	9956	9962
9948	9951	9954	9960	9966	9972	9978	9984	9990	9996			
9987	9990	9993	10000	10006	10012	10018	10024	10030	10036	10042	10048	10054
10026	10029	10032	10035	10039	10043	10049	10055	10061	10067	10073	10079	10085
10076	10079	10083	10089	10095	10101	10107	10113	10119	10125	10131	10137	10143
10138	10141	10144	10147	10150	10154	10160	10166	10172	10178	10184	10190	10196
10235	10238	10241	10244	10247	10253	10259	10265	10271	10277	10283	10289	10295
10277	10280	10283	10286	10289	10292	10295	10298	10301	10304	10307	10310	10313
10316	10319	10322	10325	10328	10331	10334	10337	10340	10343	10346	10349	10352
10356	10360	10366	10369	10375	10381	10384	10387	10390	10393	10396	10400	10406
10413	10419	10425	10428	10431	10437	10440	10446	10452	10455	10458	10461	10464

CROSS REFERENCE TABLE -- USER SYMBOLS

10467	10471	10477	10484	10490	10496	10511	10514	10519	10522	10555	10559	10561
10564	10570	10573	10578	10579	10582	10584	10588	10591	10594	10597	10599	10603
10605	10609	10612	10615	10618	10621	10624	10627	10630	10633	10637	10639	10642
10645	10648	10651	10654	10657	10660	10663	10666	10669	10672	10675	10678	10681
10684	10687	10690	10693	10696	10699	10702	10705	10708	10711	10714	10717	10720
10723	10726	10729	10732	10735	10738	10741	10744	10747	10750	10753	10756	10759
10762	10765	10768	10771	10774	10777	10780	10783	10786	10789	10792	10795	10798
10799	10802	10805	10808	10811	10814	10817	10820	10823	10826	10829	10832	10835
10838	10841	10844	10847	10850	10853	10856	10859	10862	10865	10868	10871	10874
10877	10880	10883	10886	10889	10892	10895	10898	10901	10904	10907	10910	10913
10916	10919	10922	10925	10928	10931	10934	10937	10940	10943	10946	10949	10952
10955	10958	10961	10964	10967	10970	10973	10976	10979	10982	10985	10988	10991
10994	10997	11000	11003	11006	11009	11012	11015	11018	11021	11024	11027	11030
11033	11036	11039	11042	11045	11048	11051	11054	11057	11060	11063	11066	11069
11072	11075	11078	11081	11084	11087	11090	11093	11096	11099	11102	11105	11108
11111	11114	11117	11120	11123	11126	11129	11132	11135	11138	11141	11144	11147
11150	11153	11156	11159	11162	11165	11168	11171	11174	11177	11180	11183	11186
11189	11192	11195	11198	11201	11204	11207	11210	11213	11216	11219	11222	11225
11228	11231	11234	11237	11240	11243	11246	11249	11252	11255	11258	11261	11264
11267	11270	11273	11276	11279	11282	11285	11288	11291	11294	11297	11300	11303
11306	11309	11312	11315	11318	11321	11324	11327	11330	11333	11336	11339	11342
11345	11348	11351	11354	11357	11360	11363	11366	11369	11372	11375	11378	11381
11384	11387	11390	11393	11396	11399	11402	11405	11408	11411	11414	11417	11420
11423	11426	11429	11432	11435	11438	11441	11444	11447	11450	11453	11456	11459
11462	11465	11468	11471	11474	11477	11480	11483	11486	11489	11492	11495	11498
11501	11504	11507	11510	11513	11516	11519	11522	11525	11528	11531	11534	11537
11540	11543	11546	11549	11552	11555	11558	11561	11564	11567	11570	11573	11576
11579	11582	11585	11588	11591	11594	11597	11600	11603	11606	11609	11612	11615
11618	11621	11624	11627	11630	11633	11636	11639	11642	11645	11648	11651	11654
11657	11660	11663	11666	11669	11672	11675	11678	11681	11684	11687	11690	11693
11696	11699	11702	11705	11708	11711	11714	11717	11720	11723	11726	11729	11732
11735	11738	11741	11744	11747	11750	11753	11756	11759	11762	11765	11768	11771
11774	11777	11780	11783	11786	11789	11792	11795	11798	11801	11804	11807	11810
11813	11816	11819	11822	11825	11828	11831	11834	11837	11840	11843	11846	11849
11852	11855	11858	11861	11864	11867	11870	11873	11876	11879	11882	11885	11888
11891	11894	11897	11900	11903	11906	11909	11912	11915	11918	11921	11924	11927
11930	11933	11936	11939	11942	11945	11948	11951	11954	11957	11960	11963	11966
11969	11972	11975	11978	11981	11984	11987	11990	11993	11996	11999	12002	12005
12008	12011	12014	12017	12020	12023	12026	12029	12032	12035	12038	12041	12044
12047	12050	12053	12056	12059	12062	12065	12068	12071	12074	12077	12080	12083
12086	12089	12092	12095	12098	12101	12104	12107	12110	12113	12116	12119	12122
12125	12128	12131	12134	12137	12140	12143	12146	12149	12152	12155	12158	12161
12164	12167	12170	12173	12176	12179	12182	12185	12188	12191	12194	12197	12200
12203	12206	12209	12212	12215	12218	12221	12224	12227	12230	12233	12236	12239
12242	12245	12248	12251	12254	12257	12260	12263	12266	12269	12272	12275	12278
12281	12284	12287	12290	12293	12296	12299	12302	12305	12308	12311	12314	12317
12320	12323	12326	12329	12332	12335	12338	12341	12344	12347	12350	12353	12356
12359	12362	12365	12368	12371	12374	12377	12380	12383	12386	12389	12392	12395
12398	12401	12404	12407	12410	12413	12416	12419	12422	12425	12428	12431	12434
12437	12440	12443	12446	12449	12452	12455	12458	12461	12464	12467	12470	12473
12476	12479	12482	12485	12488	12491	12494	12497	12500	12503	12506	12509	12512
12515	12518	12521	12524	12527	12530	12533	12536	12539	12542	12545	12548	12551
12554	12557	12560	12563	12566	12569	12572	12575	12578	12581	12584	12587	12590
12593	12596	12599	12602	12605	12608	12611	12614	12617	12620	12623	12626	12629
12632	12635	12638	12641	12644	12647	12650	12653	12656	12659	12662	12665	12668
12671	12674	12677	12680	12683	12686	12689	12692	12695	12698	12701	12704	12707
12710	12713	12716	12719	12722	12725	12728	12731	12734	12737	12740	12743	12746
12749	12752	12755	12758	12761	12764	12767	12770	12773	12776	12779	12782	12785
12788	12791	12794	12797	12800	12803	12806	12809	12812	12815	12818	12821	12824
12827	12830	12833	12836	12839	12842	12845	12848	12851	12854	12857	12860	12863
12866	12869	12872	12875	12878	12881	12884	12887	12890	12893	12896	12899	12902
12905	12908	12911	12914	12917	12920	12923	12926	12929	12932	12935	12938	12941
12944	12947	12950	12953	12956	12959	12962	12965	12968	12971	12974	12977	12980
12983	12986	12989	12992	12995	12998	13001	13004	13007	13010	13013	13016	13019
13022	13025	13028	13031	13034	13037	13040	13043	13046	13049	13052	13055	13058
13061	13064	13067	13070	13073	13076	13079	13082	13085	13088	13091	13094	13097
13100	13103	13106	13109	13112	13115	13118	13121	13124	13127	13130	13133	13136
13139	13142	13145	13148	13151	13154	13157	13160	13163	13166	13169	13172	13175
13178	13181	13184	13187	13190	13193	13196	13199	13202	13205	13208	13211	13214
13217	13220	13223	13226	13229	13232	13235	13238	13241	13244	13247	13250	13253
13256	13259	13262	13265	13268	13271	13274	13277	13280	13283	13286	13289	13292
13295	13298	13301	13304	13307	13310	13313	13316	13319	13322	13325	13328	13331
13334	13337	13340	13343	13346	13349	13352	13355	13358	13361	13364	13367	13370
13373	13376	13379	13382	13385	13388	13391	13394	13397	13400	13403	13406	13409
13412	13415	13418	13421	13424	13427	13430	13433	13436	13439	13442	13445	13448
13451	13454	13457	13460	13463	13466	13469	13472	13475	13478	13481	13484	13487
13490	13493	13496	13499	13502	13505	13508	13511	13514	13517	13520	13523	13526
13529	13532	13535	13538	13541	13544	13547	13550	13553	13556	13559	13562	13565
13568	13571	13574	13577	13580	13583	13586	13589	13592	13595	13598	13601	13604

CROSS REFERENCE TABLE -- USER SYMBOLS

13589	13595	13598	13601	13607	13610	13616	13622	13625	13628	13631	13634	13637
13641	13647	13654	13660	13666	13691	13714	13717	13720	13723	13726	13729	13735
13738	13742	13751	13754	13757	13767	13770	13774	13783	13785	13788	13790	13812
13822	13827	13830	13833	13836	13870	13873	13876	13879	13882	13885	13888	13892
13939	13943	13946	13949	13951	13912	13915	13918	13921	13924	13927	13930	13936
14009	14011	14014	14017	14019	13964	13967	13971	13974	13977	13980	13990	14079
14083	14085	14088	14091	14093	14046	14049	14053	14056	14059	14063	14073	14079
14151	14155	14161	14165	14171	14109	14113	14121	14125	14131	14135	14141	14145
14217	14220	14223	14226	14230	14175	14181	14186	14191	14195	14204	14207	14211
14269	14275	14278	14282	14288	14239	14243	14249	14254	14258	14265	14262	14265
14327	14330	14334	14338	14343	14291	14295	14301	14304	14308	14314	14317	14321
14380	14383	14387	14391	14395	14343	14347	14353	14356	14361	14367	14373	14376
14487	14490	14493	14497	14501	14391	14395	14401	14404	14408	14414	14418	14424
14526	14529	14533	14537	14541	14441	14445	14451	14454	14458	14464	14468	14474
14574	14580	14583	14587	14591	14491	14495	14501	14504	14508	14514	14518	14523
14641	14644	14647	14650	14654	14541	14545	14551	14554	14558	14564	14568	14574
14720	14723	14726	14729	14733	14591	14595	14601	14604	14608	14614	14618	14623
14821	14824	14827	14830	14834	14641	14645	14651	14654	14658	14664	14668	14674
14869	14872	14875	14879	14883	14691	14695	14701	14704	14708	14714	14718	14723
14933	14936	14939	14943	14947	14741	14745	14751	14754	14758	14764	14768	14774
14985	14988	14991	14994	15010	14791	14795	14801	14804	14808	14814	14818	14823
15047					14841	14845	14851	14854	14858	14864	14868	14874
2283	2285	2286	2288	2289	15013	15016	15019	15022	15025	15028	15034	15041
2314	2315	2317	2318	2319								
2337	2339	2344	2346	2347								
2379	2380	2382	2383	2384								
2407	2409	2411	2412	2413								
2449	2450	2452	2453	2454								
2483	2485	2487	2488	2489								
2503	2505	2506	2507	2508								
2549	2550	2552	2553	2554								
2576	2577	2578	2579	2580								
2603	2605	2606	2607	2608								
2644	2646	2647	2648	2649								
2696	2698	2699	2700	2701								
2725	2727	2731	2733	2734								
2770	2771	2773	2774	2775								
2798	2800	2801	2803	2804								
2832	2841	2843	2845	2846								
2891	2893	2894	2895	2896								
2920	2922	2923	2924	2925								
2955	2957	2958	2959	2960								
2994	2995	2997	2998	2999								
3027	3029	3038	3039	3040								
3088	3089	3091	3092	3093								
3119	3121	3122	3123	3124								
3150	3154	3155	3156	3157								
3187	3189	3195	3196	3197								
3216	3218	3220	3221	3222								
3269	3291	3292	3293	3294								
3314	3315	3317	3318	3319								
3345	3347	3348	3349	3350								
3376	3380	3382	3383	3384								
3413	3415	3416	3418	3419								
3445	3446	3448	3449	3451								
3478	3480	3489	3491	3522								
				3524								
				3525								
				3527								
				3529								
				3531								
				3533								
				3535								
				2312								
				2333								
				2377								
				2406								
				2447								
				2499								
				2528								
				2573								
				2601								
				2635								
				2694								
				2723								
				2768								
				2797								
				2830								
				2890								
				2918								
				2963								
				3023								
				3086								
				3118								
				3148								
				3186								
				3214								
				3256								
				3312								
				3374								
				3412								
				3443								
				3474								
				3536								

SSSSER= 00001

CROSS REFERENCE TABLE -- USER SYMBOLS

3538	3539	3540	3541	3542	3543	3544	3545	3546	3547	3548	3549	3550	3551	3552	3553	3554	3555	3556	3557	3558	3559	3560	3561	3562	3563	3564	3565
3566	3567	3568	3569	3570	3571	3572	3573	3574	3575	3576	3577	3578	3579	3580	3581	3582	3583	3584	3585	3586	3587	3588	3589	3590	3591	3592	3593
3594	3595	3596	3597	3598	3599	3600	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610	3611	3612	3613	3614	3615	3616	3617	3618	3619	3620	3621
3622	3623	3624	3625	3626	3627	3628	3629	3630	3631	3632	3633	3634	3635	3636	3637	3638	3639	3640	3641	3642	3643	3644	3645	3646	3647	3648	3649
3650	3651	3652	3653	3654	3655	3656	3657	3658	3659	3660	3661	3662	3663	3664	3665	3666	3667	3668	3669	3670	3671	3672	3673	3674	3675	3676	3677
3678	3679	3680	3681	3682	3683	3684	3685	3686	3687	3688	3689	3690	3691	3692	3693	3694	3695	3696	3697	3698	3699	3700	3701	3702	3703	3704	3705
3706	3707	3708	3709	3710	3711	3712	3713	3714	3715	3716	3717	3718	3719	3720	3721	3722	3723	3724	3725	3726	3727	3728	3729	3730	3731	3732	3733
3734	3735	3736	3737	3738	3739	3740	3741	3742	3743	3744	3745	3746	3747	3748	3749	3750	3751	3752	3753	3754	3755	3756	3757	3758	3759	3760	3761
3762	3763	3764	3765	3766	3767	3768	3769	3770	3771	3772	3773	3774	3775	3776	3777	3778	3779	3780	3781	3782	3783	3784	3785	3786	3787	3788	3789
3790	3791	3792	3793	3794	3795	3796	3797	3798	3799	3800	3801	3802	3803	3804	3805	3806	3807	3808	3809	3810	3811	3812	3813	3814	3815	3816	3817
3818	3819	3820	3821	3822	3823	3824	3825	3826	3827	3828	3829	3830	3831	3832	3833	3834	3835	3836	3837	3838	3839	3840	3841	3842	3843	3844	3845
3846	3847	3848	3849	3850	3851	3852	3853	3854	3855	3856	3857	3858	3859	3860	3861	3862	3863	3864	3865	3866	3867	3868	3869	3870	3871	3872	3873
3874	3875	3876	3877	3878	3879	3880	3881	3882	3883	3884	3885	3886	3887	3888	3889	3890	3891	3892	3893	3894	3895	3896	3897	3898	3899	3900	3901
3902	3903	3904	3905	3906	3907	3908	3909	3910	3911	3912	3913	3914	3915	3916	3917	3918	3919	3920	3921	3922	3923	3924	3925	3926	3927	3928	3929
3930	3931	3932	3933	3934	3935	3936	3937	3938	3939	3940	3941	3942	3943	3944	3945	3946	3947	3948	3949	3950	3951	3952	3953	3954	3955	3956	3957
3958	3959	3960	3961	3962	3963	3964	3965	3966	3967	3968	3969	3970	3971	3972	3973	3974	3975	3976	3977	3978	3979	3980	3981	3982	3983	3984	3985
3986	3987	3988	3989	3990	3991	3992	3993	3994	3995	3996	3997	3998	3999	4000	4001	4002	4003	4004	4005	4006	4007	4008	4009	4010	4011	4012	4013
4014	4015	4016	4017	4018	4019	4020	4021	4022	4023	4024	4025	4026	4027	4028	4029	4030	4031	4032	4033	4034	4035	4036	4037	4038	4039	4040	4041
4042	4043	4044	4045	4046	4047	4048	4049	4050	4051	4052	4053	4054	4055	4056	4057	4058	4059	4060	4061	4062	4063	4064	4065	4066	4067	4068	4069
4070	4071	4072	4073	4074	4075	4076	4077	4078	4079	4080	4081	4082	4083	4084	4085	4086	4087	4088	4089	4090	4091	4092	4093	4094	4095	4096	4097
4098	4099	4100	4101	4102	4103	4104	4105	4106	4107	4108	4109	4110	4111	4112	4113	4114	4115	4116	4117	4118	4119	4120	4121	4122	4123	4124	4125
4126	4127	4128	4129	4130	4131	4132	4133	4134	4135	4136	4137	4138	4139	4140	4141	4142	4143	4144	4145	4146	4147	4148	4149	4150	4151	4152	4153
4154	4155	4156	4157	4158	4159	4160	4161	4162	4163	4164	4165	4166	4167	4168	4169	4170	4171	4172	4173	4174	4175	4176	4177	4178	4179	4180	4181
4182	4183	4184	4185	4186	4187	4188	4189	4190	4191	4192	4193	4194	4195	4196	4197	4198	4199	4200	4201	4202	4203	4204	4205	4206	4207	4208	4209
4210	4211	4212	4213	4214	4215	4216	4217	4218	4219	4220	4221	4222	4223	4224	4225	4226	4227	4228	4229	4230	4231	4232	4233	4234	4235	4236	4237
4238	4239	4240	4241	4242	4243	4244	4245	4246	4247	4248	4249	4250	4251	4252	4253	4254	4255	4256	4257	4258	4259	4260	4261	4262	4263	4264	4265
4266	4267	4268	4269	4270	4271	4272	4273	4274	4275	4276	4277	4278	4279	4280	4281	4282	4283	4284	4285	4286	4287	4288	4289	4290	4291	4292	4293
4294	4295	4296	4297	4298	4299	4300	4301	4302	4303	4304	4305	4306	4307	4308	4309	4310	4311	4312	4313	4314	4315	4316	4317	4318	4319	4320	4321
4322	4323	4324	4325	4326	4327	4328	4329	4330	4331	4332	4333	4334	4335	4336	4337	4338	4339	4340	4341	4342	4343	4344	4345	4346	4347	4348	4349
4350	4351	4352	4353	4354	4355	4356	4357	4358	4359	4360	4361	4362	4363	4364	4365	4366	4367	4368	4369	4370	4371	4372	4373	4374	4375	4376	4377
4378	4379	4380	4381	4382	4383	4384	4385	4386	4387	4388	4389	4390	4391	4392	4393	4394	4395	4396	4397	4398	4399	4400	4401	4402	4403	4404	4405
4406	4407	4408	4409	4410	4411	4412	4413	4414	4415	4416	4417	4418	4419	4420	4421	4422	4423	4424	4425	4426	4427	4428	4429	4430	4431	4432	4433
4434	4435	4436	4437	4438	4439	4440	4441	4442	4443	4444	4445	4446	4447	4448	4449	4450	4451	4452	4453	4454	4455	4456	4457	4458	4459	4460	4461
4462	4463	4464	4465	4466	4467	4468	4469	4470	4471	4472	4473	4474	4475	4476	4477	4478	4479	4480	4481	4482	4483	4484	4485	4486	4487	4488	4489
4490	4491	4492	4493	4494	4495	4496	4497	4498	4499	4500	4501	4502	4503	4504	4505	4506	4507	4508	4509	4510	4511	4512	4513	4514	4515	4516	4517
4518	4519	4520	4521	4522	4523	4524	4525	4526	4527	4528	4529	4530	4531	4532	4533	4534	4535	4536	4537	4538	4539	4540	4541	4542	4543	4544	4545
4546	4547	4548	4549	4550	4551	4552	4553	4554	4555	4556	4557	4558	4559	4560	4561	4562	4563	4564	4565	4566	4567	4568	4569	4570	4571	4572	4573
4574	4575	4576	4577	4578	4579	4580	4581	4582	4583	4584	4585	4586	4587	4588	4589	4590	4591	4592	4593	4594	4595	4596	4597	4598	4599	4600	4601
4602	4603	4604	4605	4606	4607	4608	4609	4610	4611	4612	4613	4614	4615	4616	4617	4618	4619	4620	4621	4622	4623	4624	4625	4626	4627	4628	4629
4630	4631	4632	4633	4634	4635	4636	4637	4638	4639	4640	4641	4642	4643	4644	4645	4646	4647	4648	4649	4650	4651	4652	4653	4654	4655	4656	4657
4658	4659	4660	4661	4662	4663	4664	4665	4666	4667	4668	4669	4670	4671	4672	4673	4674	4675	4676	4677	4678	4679	4680	4681	4682	4683	4684	4685
4686	4687	4688	4689	4690	4691	4692	4693	4694	4695	4696	4697	4698	4699	4700	4701	4702	4703	4704	4705	4706	4707	4708	4709	4710	4711	4712	4713
4714	4715	4716	4717	4718	4719	4720	4721	4722	4723	4724	4725	4726	4727	4728	4729	4730	4731	4732	4733	4734	4735	4736	4737	4738	4739	4740	4741
4742	4743	4744	4745	4746	4747	4748	4749	4750	4751	4752	4753	4754	4755	4756	4757	4758	4759	4760	4761	4762	4763	4764	4765	4766	4767	4768	4769
4770	4771	4772	4773	4774	4775	4776	4777	4778	4779	4780	4781	4782	4783	4784	4785	4786	4787	4788	4789	4790	4791	4792	4793	4794	4795	4796	4797
4798	4799	4800	4801	4802	4803	4804	4805	4806	4807	4808	4809	4810	4811	4812	4813	4814	4815	4816	4817	4818	4819	4820	4821	4822	4823	4824	4825
4826	4827	4828	4829	4830	4831	4832	4833	4834	4835	4836	4837	4838	4839	4840	4841	4842	4843	4844	4845	4846	4847	4848	4849	4850	4851	4852	4853
4854	4855	4856	4857	4858	48																						

CROSS REFERENCE TABLE -- USER SYMBOLS

5708	5709	5711	5721	5723	5724	5726	5727	5729	5730	5732	5733	5735
5736	5738	5739	5741	5743	5745	5749	5751	5756	5758	5762	5764	5768
5770	5782	5784	5785	5787	5789	5790	5791	5793	5794	5796	5806	5808
5809	5811	5812	5814	5815	5817	5818	5820	5821	5823	5824	5826	5828
5830	5831	5832	5834	5835	5837	5839	5840	5841	5843	5844	5846	5848
5850	5851	5852	5854	5855	5857	5859	5860	5861	5863	5864	5866	5868
5870	5871	5872	5874	5875	5877	5879	5880	5881	5883	5884	5886	5888
5890	5891	5892	5894	5895	5897	5899	5900	5901	5903	5904	5906	5908
5910	5911	5912	5914	5915	5917	5919	5920	5921	5923	5924	5926	5928
5930	5931	5932	5934	5935	5937	5939	5940	5941	5943	5944	5946	5948
5950	5951	5952	5954	5955	5957	5959	5960	5961	5963	5964	5966	5968
5970	5971	5972	5974	5975	5977	5979	5980	5981	5983	5984	5986	5988
5990	5991	5992	5994	5995	5997	5999	6000	6001	6003	6004	6006	6008
6010	6011	6012	6014	6015	6017	6019	6020	6021	6023	6024	6026	6028
6030	6031	6032	6034	6035	6037	6039	6040	6041	6043	6044	6046	6048
6050	6051	6052	6054	6055	6057	6059	6060	6061	6063	6064	6066	6068
6070	6071	6072	6074	6075	6077	6079	6080	6081	6083	6084	6086	6088
6090	6091	6092	6094	6095	6097	6099	6100	6101	6103	6104	6106	6108
6110	6111	6112	6114	6115	6117	6119	6120	6121	6123	6124	6126	6128
6130	6131	6132	6134	6135	6137	6139	6140	6141	6143	6144	6146	6148
6150	6151	6152	6154	6155	6157	6159	6160	6161	6163	6164	6166	6168
6170	6171	6172	6174	6175	6177	6179	6180	6181	6183	6184	6186	6188
6190	6191	6192	6194	6195	6197	6199	6200	6201	6203	6204	6206	6208
6210	6211	6212	6214	6215	6217	6219	6220	6221	6223	6224	6226	6228
6230	6231	6232	6234	6235	6237	6239	6240	6241	6243	6244	6246	6248
6250	6251	6252	6254	6255	6257	6259	6260	6261	6263	6264	6266	6268
6270	6271	6272	6274	6275	6277	6279	6280	6281	6283	6284	6286	6288
6290	6291	6292	6294	6295	6297	6299	6300	6301	6303	6304	6306	6308
6310	6311	6312	6314	6315	6317	6319	6320	6321	6323	6324	6326	6328
6330	6331	6332	6334	6335	6337	6339	6340	6341	6343	6344	6346	6348
6350	6351	6352	6354	6355	6357	6359	6360	6361	6363	6364	6366	6368
6370	6371	6372	6374	6375	6377	6379	6380	6381	6383	6384	6386	6388
6390	6391	6392	6394	6395	6397	6399	6400	6401	6403	6404	6406	6408
6410	6411	6412	6414	6415	6417	6419	6420	6421	6423	6424	6426	6428
6430	6431	6432	6434	6435	6437	6439	6440	6441	6443	6444	6446	6448
6450	6451	6452	6454	6455	6457	6459	6460	6461	6463	6464	6466	6468
6470	6471	6472	6474	6475	6477	6479	6480	6481	6483	6484	6486	6488
6490	6491	6492	6494	6495	6497	6499	6500	6501	6503	6504	6506	6508
6510	6511	6512	6514	6515	6517	6519	6520	6521	6523	6524	6526	6528
6530	6531	6532	6534	6535	6537	6539	6540	6541	6543	6544	6546	6548
6550	6551	6552	6554	6555	6557	6559	6560	6561	6563	6564	6566	6568
6570	6571	6572	6574	6575	6577	6579	6580	6581	6583	6584	6586	6588
6590	6591	6592	6594	6595	6597	6599	6600	6601	6603	6604	6606	6608
6610	6611	6612	6614	6615	6617	6619	6620	6621	6623	6624	6626	6628
6630	6631	6632	6634	6635	6637	6639	6640	6641	6643	6644	6646	6648
6650	6651	6652	6654	6655	6657	6659	6660	6661	6663	6664	6666	6668
6670	6671	6672	6674	6675	6677	6679	6680	6681	6683	6684	6686	6688
6690	6691	6692	6694	6695	6697	6699	6700	6701	6703	6704	6706	6708
6710	6711	6712	6714	6715	6717	6719	6720	6721	6723	6724	6726	6728
6730	6731	6732	6734	6735	6737	6739	6740	6741	6743	6744	6746	6748
6750	6751	6752	6754	6755	6757	6759	6760	6761	6763	6764	6766	6768
6770	6771	6772	6774	6775	6777	6779	6780	6781	6783	6784	6786	6788
6790	6791	6792	6794	6795	6797	6799	6800	6801	6803	6804	6806	6808
6810	6811	6812	6814	6815	6817	6819	6820	6821	6823	6824	6826	6828
6830	6831	6832	6834	6835	6837	6839	6840	6841	6843	6844	6846	6848
6850	6851	6852	6854	6855	6857	6859	6860	6861	6863	6864	6866	6868
6870	6871	6872	6874	6875	6877	6879	6880	6881	6883	6884	6886	6888
6890	6891	6892	6894	6895	6897	6899	6900	6901	6903	6904	6906	6908
6910	6911	6912	6914	6915	6917	6919	6920	6921	6923	6924	6926	6928
6930	6931	6932	6934	6935	6937	6939	6940	6941	6943	6944	6946	6948
6950	6951	6952	6954	6955	6957	6959	6960	6961	6963	6964	6966	6968
6970	6971	6972	6974	6975	6977	6979	6980	6981	6983	6984	6986	6988
6990	6991	6992	6994	6995	6997	6999	7000	7001	7003	7004	7006	7008
7010	7011	7012	7014	7015	7017	7019	7020	7021	7023	7024	7026	7028
7030	7031	7032	7034	7035	7037	7039	7040	7041	7043	7044	7046	7048
7050	7051	7052	7054	7055	7057	7059	7060	7061	7063	7064	7066	7068
7070	7071	7072	7074	7075	7077	7079	7080	7081	7083	7084	7086	7088
7090	7091	7092	7094	7095	7097	7099	7100	7101	7103	7104	7106	7108
7110	7111	7112	7114	7115	7117	7119	7120	7121	7123	7124	7126	7128
7130	7131	7132	7134	7135	7137	7139	7140	7141	7143	7144	7146	7148
7150	7151	7152	7154	7155	7157	7159	7160	7161	7163	7164	7166	7168
7170	7171	7172	7174	7175	7177	7179	7180	7181	7183	7184	7186	7188
7188	7189	7190	7192	7194	7198	7200	7205	7207	7211	7212	7214	7216
7233	7234	7236	7237	7239	7240	7242	7243	7245	7249	7251	7253	7255
7261	7263	7264	7266	7267	7269	7270	7272	7273	7275	7277	7279	7281
7285	7290	7292	7296	7298	7302	7304	7313	7315	7319	7321	7323	7325
7350	7352	7353	7355	7356	7358	7359	7361	7362	7365	7367	7369	7371
7370	7374	7376	7377	7379	7380	7382	7383	7385	7389	7391	7393	7395
7392	7394	7395	7397	7398	7400	7401	7403	7404	7406	7407	7409	7411
7415	7425	7427	7428	7430	7431	7433	7434	7436	7437	7439	7441	7443
7450	7452	7462	7464	7465	7467	7468	7470	7471	7473	7475	7477	7479
7487	7518	7520	7521	7523	7524	7526	7527	7529	7530	7532	7533	7535
7536	7538	7539	7541	7542	7544	7545	7547	7548	7550	7551	7553	7555
7559	7560	7562	7566	7568	7569	7571	7572	7574	7575	7577	7578	7580
7581	7583	7584	7586	7587	7589	7590	7592	7593	7595	7596	7598	7599
7601	7602	7604	7605	7607	7608	7610	7614	7616	7623	7625	7626	7628
7629	7631	7632	7634	7635	7637	7638	7640	7641	7643	7647	7649	7650

CROSS REFERENCE TABLE -- USER SYMBOLS

7652	7662	7664	7665	7667	7668	7670	7671	7673	7674	7676	7678	7680
7687	7689	7690	7692	7702	7704	7705	7707	7708	7710	7711	7713	7714
7716	7718	7720	7728	7730	7761	7763	7764	7766	7767	7769	7770	7772
7773	7775	7776	7778	7779	7781	7782	7784	7785	7787	7788	7790	7791
7793	7794	7796	7797	7799	7800	7802	7803	7805	7806	7808	7809	7811
7812	7814	7818	7820	7821	7823	7824	7826	7839	7841	7842	7844	7846
7848	7855	7857	7858	7860	7865	7867	7898	7900	7901	7903	7904	7906
7907	7909	7910	7912	7913	7915	7916	7918	7919	7921	7922	7924	7925
7927	7928	7930	7931	7933	7934	7936	7937	7939	7940	7942	7943	7945
7946	7948	7951	7953	7954	7956	7957	7959	7970	7972	7973	7975	7976
7978	7979	7981	7982	7984	7985	7987	7991	7993	7994	7996	8015	8017
8018	8020	8021	8023	8024	8026	8027	8029	8030	8032	8033	8035	8036
8038	8039	8041	8042	8044	8045	8047	8048	8050	8050	8052	8063	8065
8066	8068	8069	8071	8072	8074	8076	8078	8104	8106	8139	8141	8142
8144	8145	8147	8148	8150	8151	8153	8154	8156	8157	8159	8160	8162
8164	8166	8167	8169	8170	8172	8182	8184	8186	8187	8188	8190	8191
8193	8194	8196	8197	8199	8201	8203	8207	8209	8214	8216	8220	8222
8226	8228	8241	8243	8254	8257	8259	8259	8264	8266	8296	8298	8299
8301	8302	8304	8305	8307	8308	8310	8311	8313	8314	8316	8317	8319
8321	8323	8324	8326	8327	8329	8339	8341	8342	8344	8345	8347	8348
8350	8351	8353	8354	8356	8358	8360	8364	8366	8371	8373	8377	8379
8383	8385	8401	8403	8414	8416	8417	8419	8424	8426	8456	8458	8459
8461	8462	8464	8465	8467	8468	8470	8471	8473	8474	8476	8477	8479
8489	8491	8492	8494	8495	8497	8498	8500	8501	8503	8504	8506	8507
8509	8511	8513	8521	8523	8524	8528	8531	8532	8535	8537	8538	8540
8561	8563	8564	8566	8567	8569	8573	8574	8575	8577	8578	8587	8588
8590	8591	8593	8595	8597	8605	8607	8620	8622	8637	8639	8669	8671
8672	8674	8675	8677	8678	8680	8681	8683	8684	8686	8687	8689	8690
8692	8712	8714	8715	8717	8718	8720	8721	8723	8724	8726	8727	8729
8739	8741	8742	8744	8745	8747	8748	8750	8751	8753	8755	8757	8761
8763	8768	8770	8774	8776	8786	8788	8803	8805	8835	8837	8838	8840
8841	8843	8844	8846	8847	8849	8850	8852	8853	8855	8856	8858	8859
8861	8868	8870	8875	8877	8878	8880	8882	8884	8892	8894	8904	8906
8907	8909	8919	8921	8926	8928	8931	8933	8934	8936	8937	8939	8970
8972	8973	8975	8976	8978	8979	8981	8985	8987	8988	8990	8991	8993
8994	8996	8997	8999	9000	9002	9003	9005	9006	9008	9009	9011	9012
9014	9015	9017	9018	9020	9021	9023	9024	9026	9027	9029	9030	9032
9033	9035	9036	9038	9039	9041	9042	9044	9045	9047	9048	9050	9051
9053	9054	9056	9057	9059	9060	9062	9063	9065	9066	9068	9069	9071
9072	9074	9075	9077	9078	9080	9081	9083	9084	9086	9088	9090	9092
9094	9098	9100	9101	9103	9113	9115	9116	9118	9119	9121	9122	9124
9125	9127	9128	9130	9132	9134	9138	9140	9145	9147	9151	9153	9157
9159	9160	9162	9163	9165	9169	9171	9172	9174	9184	9186	9187	9189
9190	9192	9193	9195	9196	9198	9199	9201	9203	9205	9209	9211	9216
9218	9222	9224	9228	9230	9233	9243	9245	9248	9251	9253	9254	9256
9287	9289	9290	9292	9293	9295	9296	9298	9302	9304	9305	9307	9308
9310	9311	9313	9314	9316	9317	9319	9320	9322	9323	9325	9326	9328
9329	9331	9332	9334	9335	9337	9338	9340	9341	9343	9344	9346	9347
9349	9350	9352	9353	9355	9356	9358	9359	9361	9362	9364	9365	9367
9368	9370	9371	9373	9374	9376	9377	9379	9380	9382	9383	9385	9386
9388	9389	9391	9392	9394	9395	9397	9398	9400	9401	9403	9405	9407
9409	9411	9415	9417	9430	9432	9433	9435	9436	9438	9439	9441	9442
9444	9445	9447	9449	9451	9455	9457	9462	9464	9468	9470	9474	9476
9477	9479	9480	9482	9489	9491	9501	9503	9504	9506	9507	9509	9510
9512	9513	9515	9516	9518	9520	9522	9526	9528	9533	9535	9539	9541
9545	9547	9560	9562	9595	9597	9598	9600	9601	9603	9604	9606	9607

H10

CROSS REFERENCE TABLE -- USER SYMBOLS

9609	9610	9612	9613	9615	9619	9621	9622	9624	9625	9627	9628	9630
9631	9633	9634	9635	9637	9639	9640	9642	9643	9645	9646	9648	9649
9651	9652	9654	9655	9657	9659	9660	9662	9663	9664	9666	9667	9669
9670	9672	9673	9674	9675	9677	9678	9679	9680	9681	9682	9683	9684
9690	9691	9692	9693	9694	9695	9696	9697	9698	9699	9700	9701	9702
9703	9704	9705	9706	9707	9708	9709	9710	9711	9712	9713	9714	9715
9716	9717	9718	9719	9720	9721	9722	9723	9724	9725	9726	9727	9728
9729	9730	9731	9732	9733	9734	9735	9736	9737	9738	9739	9740	9741
9742	9743	9744	9745	9746	9747	9748	9749	9750	9751	9752	9753	9754
9755	9756	9757	9758	9759	9760	9761	9762	9763	9764	9765	9766	9767
9768	9769	9770	9771	9772	9773	9774	9775	9776	9777	9778	9779	9780
9781	9782	9783	9784	9785	9786	9787	9788	9789	9790	9791	9792	9793
9794	9795	9796	9797	9798	9799	9800	9801	9802	9803	9804	9805	9806
9807	9808	9809	9810	9811	9812	9813	9814	9815	9816	9817	9818	9819
9820	9821	9822	9823	9824	9825	9826	9827	9828	9829	9830	9831	9832
9833	9834	9835	9836	9837	9838	9839	9840	9841	9842	9843	9844	9845
9846	9847	9848	9849	9850	9851	9852	9853	9854	9855	9856	9857	9858
9859	9860	9861	9862	9863	9864	9865	9866	9867	9868	9869	9870	9871
9872	9873	9874	9875	9876	9877	9878	9879	9880	9881	9882	9883	9884
9885	9886	9887	9888	9889	9890	9891	9892	9893	9894	9895	9896	9897
9898	9899	9900	9901	9902	9903	9904	9905	9906	9907	9908	9909	9910
10011	10013	10014	10016	10017	10019	10020	10022	10023	10025	10027	10028	10010
10031	10032	10034	10036	10037	10039	10040	10042	10043	10045	10047	10048	10029
10067	10069	10070	10072	10073	10075	10076	10078	10079	10081	10083	10084	10066
10091	10092	10093	10094	10095	10096	10097	10098	10099	10101	10102	10103	10089
10135	10137	10138	10140	10141	10143	10144	10146	10147	10149	10150	10151	10125
10156	10160	10162	10167	10169	10173	10175	10179	10181	10194	10196	10197	10154
10232	10234	10235	10237	10238	10240	10241	10243	10244	10246	10247	10249	10231
10255	10256	10258	10259	10261	10264	10265	10267	10268	10270	10271	10272	10253
10274	10276	10277	10279	10280	10281	10282	10283	10284	10286	10287	10288	10273
10294	10295	10297	10299	10300	10301	10302	10303	10304	10306	10307	10308	10292
10313	10315	10316	10318	10319	10320	10321	10322	10323	10325	10327	10328	10312
10333	10334	10336	10337	10339	10340	10341	10343	10345	10346	10348	10349	10331
10352	10354	10356	10358	10359	10360	10361	10362	10363	10365	10366	10367	10351
10389	10390	10392	10393	10394	10395	10396	10397	10398	10399	10400	10401	10387
10419	10421	10422	10427	10428	10430	10431	10433	10440	10442	10448	10449	10415
10457	10458	10460	10461	10463	10464	10466	10467	10469	10471	10473	10474	10455
10484	10486	10490	10492	10493	10494	10495	10496	10497	10498	10499	10500	10479
10554	10555	10557	10558	10560	10561	10563	10564	10566	10568	10569	10570	10552
10576	10578	10579	10581	10582	10584	10585	10587	10588	10590	10591	10592	10575
10596	10597	10599	10600	10602	10603	10605	10606	10608	10609	10611	10612	10594
10615	10617	10618	10620	10621	10623	10624	10626	10627	10629	10630	10632	10614
10635	10636	10638	10639	10641	10642	10644	10645	10647	10648	10650	10652	10637
10654	10656	10657	10659	10660	10662	10663	10665	10666	10668	10669	10671	10653
10675	10677	10679	10683	10685	10686	10688	10689	10690	10692	10693	10694	10673
10710	10712	10713	10715	10717	10719	10720	10723	10725	10730	10734	10735	10709
10744	10745	10747	10748	10750	10751	10752	10753	10755	10759	10761	10762	10742
10778	10780	10781	10783	10784	10786	10788	10789	10790	10791	10792	10793	10777
10809	10813	10815	10828	10830	10833	10835	10836	10838	10839	10840	10841	10807
10875	10877	10878	10880	10881	10883	10887	10889	10890	10892	10893	10894	10874
10898	10899	10901	10902	10904	10905	10907	10908	10909	10910	10911	10912	10896
10917	10919	10920	10922	10923	10925	10926	10928	10929	10931	10932	10933	10916
10937	10938	10940	10941	10943	10944	10946	10947	10949	10950	10952	10953	10935
10956	10958	10959	10961	10962	10964	10965	10967	10968	10970	10971	10973	10955
10976	10977	10979	10980	10982	10983	10985	10986	10988	10990	10992	10994	10974
11000	11002	11015	11017	11018	11020	11021	11023	11024	11026	11027	11029	10996
11032	11034	11036	11040	11042	11047	11049	11053	11055	11059	11061	11062	11030
11065	11067	11074	11076	11086	11088	11089	11091	11092	11094	11095	11097	11064
11100	11101	11103	11105	11107	11111	11113	11118	11120	11124	11126	11130	11098
11145	11147	11180	11182	11183	11185	11186	11188	11189	11191	11192	11194	11132
11197	11198	11200	11204	11206	11207	11209	11210	11212	11213	11215	11216	11195
11219	11221	11222	11224	11225	11227	11228	11230	11231	11233	11234	11236	11218

CROSS REFERENCE TABLE -- USER SYMBOLS

12866	12867	12869	12870	12872	12873	12875	12876	12878	12879	12881	12882	12884
12865	12866	12868	12869	12871	12872	12874	12875	12877	12878	12880	12881	12883
12892	12893	12895	12896	12897	12898	12899	12900	12901	12902	12903	12904	12905
12922	12923	12925	12926	12927	12928	12929	12930	12931	12932	12933	12934	12935
12955	12956	12958	12959	12960	12961	12962	12963	12964	12965	12966	12967	12968
13013	13014	13016	13017	13018	13019	13020	13021	13022	13023	13024	13025	13026
13105	13106	13108	13109	13110	13111	13112	13113	13114	13115	13116	13117	13118
13125	13126	13128	13129	13130	13131	13132	13133	13134	13135	13136	13137	13138
13168	13169	13171	13172	13173	13174	13175	13176	13177	13178	13179	13180	13181
13187	13188	13190	13191	13192	13193	13194	13195	13196	13197	13198	13199	13200
13207	13208	13210	13211	13212	13213	13214	13215	13216	13217	13218	13219	13220
13274	13275	13277	13278	13279	13280	13281	13282	13283	13284	13285	13286	13287
13311	13312	13314	13315	13316	13317	13318	13319	13320	13321	13322	13323	13324
13339	13340	13342	13343	13344	13345	13346	13347	13348	13349	13350	13351	13352
13408	13409	13411	13412	13413	13414	13415	13416	13417	13418	13419	13420	13421
13431	13432	13434	13435	13436	13437	13438	13439	13440	13441	13442	13443	13444
13450	13451	13453	13454	13455	13456	13457	13458	13459	13460	13461	13462	13463
13470	13471	13473	13474	13475	13476	13477	13478	13479	13480	13481	13482	13483
13499	13500	13502	13503	13504	13505	13506	13507	13508	13509	13510	13511	13512
13509	13510	13512	13513	13514	13515	13516	13517	13518	13519	13520	13521	13522
13530	13531	13533	13534	13535	13536	13537	13538	13539	13540	13541	13542	13543
13555	13556	13558	13559	13560	13561	13562	13563	13564	13565	13566	13567	13568
13583	13584	13586	13587	13588	13589	13590	13591	13592	13593	13594	13595	13596
13623	13624	13626	13627	13628	13629	13630	13631	13632	13633	13634	13635	13636
13651	13652	13654	13655	13656	13657	13658	13659	13660	13661	13662	13663	13664
13678	13679	13681	13682	13683	13684	13685	13686	13687	13688	13689	13690	13691
13728	13729	13731	13732	13733	13734	13735	13736	13737	13738	13739	13740	13741
13757	13758	13760	13761	13762	13763	13764	13765	13766	13767	13768	13769	13770
13807	13808	13810	13811	13812	13813	13814	13815	13816	13817	13818	13819	13820
13827	13828	13830	13831	13832	13833	13834	13835	13836	13837	13838	13839	13840
13909	13910	13912	13913	13914	13915	13916	13917	13918	13919	13920	13921	13922
13932	13933	13935	13936	13937	13938	13939	13940	13941	13942	13943	13944	13945
13951	13952	13954	13955	13956	13957	13958	13959	13960	13961	13962	13963	13964
13992	13993	13995	13996	13997	13998	13999	14000	14001	14002	14003	14004	14005
14046	14047	14049	14050	14051	14052	14053	14054	14055	14056	14057	14058	14059
14075	14076	14078	14079	14080	14081	14082	14083	14084	14085	14086	14087	14088
14103	14104	14106	14107	14108	14109	14110	14111	14112	14113	14114	14115	14116
14143	14144	14146	14147	14148	14149	14150	14151	14152	14153	14154	14155	14156
14175	14176	14178	14179	14180	14181	14182	14183	14184	14185	14186	14187	14188
14209	14210	14212	14213	14214	14215	14216	14217	14218	14219	14220	14221	14222
14236	14237	14239	14240	14241	14242	14243	14244	14245	14246	14247	14248	14249
14254	14255	14257	14258	14259	14260	14261	14262	14263	14264	14265	14266	14267
14281	14282	14284	14285	14286	14287	14288	14289	14290	14291	14292	14293	14294
14319	14320	14322	14323	14324	14325	14326	14327	14328	14329	14330	14331	14332
14349	14350	14352	14353	14354	14355	14356	14357	14358	14359	14360	14361	14362
14378	14379	14381	14382	14383	14384	14385	14386	14387	14388	14389	14390	14391
14428	14429	14431	14432	14433	14434	14435	14436	14437	14438	14439	14440	14441
14459	14460	14462	14463	14464	14465	14466	14467	14468	14469	14470	14471	14472
14489	14490	14492	14493	14494	14495	14496	14497	14498	14499	14500	14501	14502
14529	14530	14532	14533	14534	14535	14536	14537	14538	14539	14540	14541	14542
14569	14570	14572	14573	14574	14575	14576	14577	14578	14579	14580	14581	14582
14591	14592	14594	14595	14596	14597	14598	14599	14600	14601	14602	14603	14604
14598	14599	14601	14602	14603	14604	14605	14606	14607	14608	14609	14610	14611
14640	14641	14643	14644	14645	14646	14647	14648	14649	14650	14651	14652	14653
14640	14641	14643	14644	14645	14646	14647	14648	14649	14650	14651	14652	14653

CROSS REFERENCE TABLE -- USER SYMBOLS

14655	14654	14656	14659	14701	14702	14703	14705	14707	14708	14710	14711	14713
14714	14716	14717	14718	14723	14725	14726	14728	14732	14734	14738	14740	14747
14749	14750	14752	14758	14760	14784	14786	14815	14817	14818	14820	14821	14823
14824	14825	14830	14838	14833	14835	14839	14841	14845	14847	14851	14853	14860
14862	14865	14868	14875	14871	14872	14874	14891	14893	14894	14896	14897	14899
14900	14902	14903	14905	14909	14911	14915	14917	14921	14923	14930	14932	14936
14938	14939	14941	14942	14944	14955	14957	14958	14960	14961	14963	14964	14966
14967	14969	14973	14975	14979	14981	14985	14987	14988	14990	14994	14996	15010
15012	15013	15015	15016	15018	15019	15021	15022	15024	15025	15027	15028	15030
15034	15036	15041	15043	15047	15049							
1046												
203	205	208	221	222	224	226	228	232	236	246	294	380
381	382	383	392	480	486	487	489	491	534	536	537	538
539	541	542	543	544	546	547	548	549	551	552	553	554
556	557	558	559	561	562	563	564	566	567	568	569	571
572	573	574	576	577	578	579	581	582	583	584	586	587
588	589	591	592	593	594	596	597	598	599	601	602	603
604	606	607	608	609	611	612	613	614	775	785	892	905
1010	1081	1082	1174	1226	1236	1328	1329	1335	1389	1678	1707	1750
1769	1785	1787	1841	1844	1865	1864	2005	2088	2092	2138	2169	2202
2205	2281	2478	2675	2872	3070	3287	3520	3717	3914	4111	4308	4505
4702	4899	5097	5274	5481	5688	5895	6102	6309	6516	6723	6930	7137
7343	7517	7760	7897	8137	8294	8455	8668	8834	8960	9277	9594	9911
10228	10545	10862	11179	11496	11813	12130	12447	12764	13081	13398	13713	13856
14480	14698	14808	15310	15312								
1	5162	5232	8009	8097	8238	8398	8533	8617	8696	8783	8865	8916
12905	12973	13792	13968	14770	14773							
1	9095	9166	9231	9412	9483	9548	9729	9800	9865	10046	10117	10182
10363	10434	10499	10680	10686	10751	10754	10816	10997	11068	11133	11314	11385
11450	11631	11702	11767	11948	12019	12084	12265	12336	12401	12582	12653	12718
12899	12970	13035	13216	13287	13352	13533	13604	13669	14863	14933		
1	11637	11705										
1429	1569											
1	3092	3181	3309	3407	7861	9735	9803	14535	14836	14906	14970	14991
1	7410	7617	7644	7827	7988	8871	8922	14571	14854	15037		
1	14577	14592	14924									
1	13732	13760	13786	14559	14741							
1	2362	2442	2559	2639	2756	2836	2953	3033	3160	3249	3386	3484
3601	3681	3798	3878	3995	4075	4192	4272	4389	4469	4586	4666	4783
4863	4980	5060	5168	5238	5360	5445	5567	5652	5774	5859	5981	6066
6188	6273	6395	6480	6602	6687	6809	6894	7016	7101	7223	7308	8232
8389	13798	14547	14735	14761	14848	14857	14918	14927	14982	14997	15031	
1	7833	8250	8410	8545	8629	8705	8795	13983	14015	14392	14776	
920												
1												
2537	2334	2340	2347	2353	2365	2414	2420	2427	2433	2450	2453	2531
2747	2544	2550	2562	2611	2617	2624	2630	2647	2650	2728	2734	2741
3005	2759	2808	2814	2821	2827	2844	2947	2925	2931	2938	2944	2956
3234	3011	3018	3024	3041	3044	3132	3138	3145	3151	3163	3221	3227
3492	3240	3257	3260	3358	3364	3371	3377	3389	3456	3462	3469	3475
3770	3495	3573	3579	3586	3592	3604	3653	3659	3666	3672	3689	3692
3980	3776	3783	3789	3801	3850	3856	3863	3869	3886	3889	3967	3973
4195	3986	3998	4047	4053	4060	4066	4083	4086	4164	4170	4177	4183
4447	4244	4250	4257	4263	4280	4283	4361	4367	4374	4380	4392	4441
4657	4454	4460	4477	4480	4558	4564	4571	4577	4589	4638	4644	4651
4874	4674	4677	4755	4761	4768	4774	4786	4835	4841	4848	4854	4871
	4952	4958	4965	4971	4983	5032	5038	5045	5051	5068	5071	5140

S40CAT: ***** U
 = 037470

.ADC = 000100

.ADD = 000000

.ADDMC= 000020

.ADVAM 006050

.AND = 000260

.B80 = 002000

.B81 = 002400

.B84 = 003000

.B87 = 003400

.BC = 001000

.BEGIN 003464

.BR = 000400

CROSS REFERENCE TABLE -- USER SYMBOLS

5143	5150	5153	5171	5210	5213	5220	5223	5234	5249	5333	5338	5345
5351	5363	5417	5423	5430	5436	5453	5459	5462	5475	5533	5538	5570
5824	5830	5837	5843	5850	5856	5862	5868	5874	5881	5933	5938	5970
5844	5850	5857	5863	5870	5876	5882	5888	5894	5901	5953	5958	5990
6074	6077	6160	6166	6173	6179	6191	6197	6204	6211	6251	6256	6284
6367	6373	6380	6386	6393	6400	6406	6412	6419	6425	6471	6476	6500
6587	6593	6605	6611	6618	6624	6630	6636	6642	6648	6694	6699	6723
6812	6816	6822	6829	6835	6841	6847	6853	6859	6865	6911	6916	6940
7079	7086	7092	7099	7105	7111	7117	7123	7129	7135	7181	7186	7210
7299	7316	7319	7344	7347	7353	7359	7365	7371	7377	7423	7428	7452
7731	7734	7836	7849	7852	7858	7864	7870	7876	7882	7928	7933	7957
8110	8204	8210	8217	8223	8229	8235	8241	8247	8253	8300	8305	8329
8420	8427	8430	8437	8443	8449	8455	8461	8467	8473	8520	8525	8549
8640	8643	8708	8714	8720	8726	8732	8738	8744	8750	8796	8801	8825
8898	8923	8933	8940	8946	8952	8958	8964	8970	8976	9022	9027	9051
9249	9253	9258	9141	9147	9153	9159	9165	9171	9177	9224	9229	9253
9775	9782	9788	9846	9849	9855	9861	9867	9873	9879	9926	9931	9955
10105	10157	10163	10170	10176	10182	10188	10194	10200	10206	10253	10258	10282
10480	10487	10493	10507	10514	10520	10526	10532	10538	10544	10591	10596	10620
10810	10824	10831	10834	10837	10843	10849	10855	10861	10867	10914	10919	10943
11148	11151	11154	11160	11167	11173	11179	11185	11191	11197	11244	11249	11273
11671	11677	11684	11690	11742	11748	11754	11760	11766	11772	11819	11824	11848
12001	12007	12053	12056	12072	12079	12085	12091	12097	12103	12150	12155	12179
12376	12389	12393	12396	12402	12408	12414	12420	12426	12432	12479	12484	12508
12706	12712	12725	12731	12737	12743	12749	12755	12761	12767	12814	12819	12843
13043	13050	13053	13056	13062	13068	13074	13080	13086	13092	13139	13144	13168
13370	13373	13379	13385	13391	13397	13403	13409	13415	13421	13468	13473	13497
13748	13763	13777	13780	13786	13792	13798	13804	13810	13816	13863	13868	13892
14024	14040	14056	14066	14076	14086	14096	14106	14116	14126	14173	14178	14202
14168	14178	14188	14198	14204	14210	14216	14222	14228	14234	14281	14286	14310
14337	14383	14386	14396	14404	14414	14424	14434	14444	14454	14501	14506	14530
14660	14667	14670	14764	14779	14787	14797	14807	14817	14827	14874	14879	14903
18	2310	2333	2341	2348	2354	2360	2366	2372	2378	2425	2430	2454
2443	2451	2454	2461	2467	2473	2479	2485	2491	2497	2544	2549	2573
2809	2815	2822	2828	2834	2840	2846	2852	2858	2864	2911	2916	2940
2957	2981	3006	3012	3019	3025	3031	3037	3043	3049	3096	3101	3125
3152	3161	3164	3171	3177	3183	3189	3195	3201	3207	3254	3259	3283
3365	3372	3378	3387	3393	3399	3405	3411	3417	3423	3470	3475	3499
3549	3574	3580	3587	3593	3599	3605	3611	3617	3623	3670	3675	3699
3690	3693	3746	3771	3790	3800	3810	3820	3830	3840	3887	3892	3916
3970	3879	3887	3890	3893	3899	3905	3911	3917	3923	3970	3975	3999
4054	4061	4067	4076	4084	4094	4104	4114	4124	4134	4181	4186	4210
4220	4245	4251	4258	4264	4270	4276	4282	4288	4294	4341	4346	4370
4390	4393	4417	4442	4448	4454	4460	4466	4472	4478	4525	4530	4554
4572	4578	4587	4590	4596	4604	4614	4624	4634	4644	4691	4696	4720
4756	4762	4769	4775	4784	4794	4804	4814	4824	4834	4881	4886	4910
4875	4888	4893	4899	4904	4910	4916	4922	4928	4934	4981	4986	5010
5061	5069	5072	5075	5081	5087	5093	5099	5105	5111	5158	5163	5187
5221	5224	5239	5247	5250	5256	5264	5274	5284	5294	5341	5346	5370
5418	5424	5431	5437	5446	5454	5464	5474	5484	5494	5541	5546	5570
5571	5597	5625	5631	5638	5644	5654	5664	5674	5684	5731	5736	5760
5766	5775	5778	5804	5822	5838	5845	5851	5857	5863	5910	5915	5939
5960	5967	5973	5982	5985	6011	6039	6045	6051	6057	6104	6109	6133
6133	6161	6167	6174	6180	6189	6192	6218	6246	6252	6299	6304	6328

BSRC= 160000
BSYH= 100000

CROSS REFERENCE TABLE -- USER SYMBOLS

6282	6285	6340	6368	6374	6381	6387	6396	6399	6425	6453	6459	6466
6472	6481	6489	6498	6547	6575	6581	6588	6594	6603	6606	6632	6660
6666	6673	6679	6688	6698	6699	6754	6782	6788	6795	6801	6810	6813
6839	6867	6873	6880	6889	6897	6903	6906	6961	6989	6995	7002	7008
7017	7020	7046	7074	7080	7087	7093	7102	7110	7113	7168	7196	7202
7209	7215	7224	7257	7287	7288	7287	7294	7300	7309	7317	7320	7411
7423	7445	7448	7460	7479	7482	7489	7492	7618	7645	7660	7682	7685
7700	7722	7725	7732	7736	7828	7834	7837	7850	7853	7869	7872	7876
7959	7989	8001	8013	8056	8080	8083	8089	8101	8108	8111	8180	8205
8211	8218	8224	8233	8251	8261	8268	8271	8337	8362	8368	8375	8381
8390	8393	8411	8421	8431	8431	8487	8515	8518	8525	8528	8546	8549
8577	8579	8602	8609	8612	8630	8633	8641	8644	8706	8709	8737	8759
8765	8772	8778	8799	8812	8807	8810	8872	8886	8889	8896	8899	8923
8930	8933	9111	9136	9143	9149	9155	9182	9207	9213	9220	9226	9237
9240	9247	9250	9256	9257	9259	9266	9275	9499	9524	9530	9537	9543
9554	9557	9564	9579	9584	9599	9776	9783	9789	9816	9841	9847	9854
9860	9871	9874	9881	9884	10062	10087	10093	10100	10106	10133	10158	10164
10171	10177	10188	10191	10198	10201	10207	10404	10410	10417	10423	10450	10475
10481	10488	10494	10505	10508	10515	10518	10526	10721	10727	10734	10740	10767
10792	10798	10805	10811	10828	10839	10832	10836	11013	11038	11044	11051	11057
11084	11109	11115	11122	11128	11139	11142	11149	11152	11330	11335	11361	11368
11374	11401	11426	11432	11438	11455	11458	11459	11466	11469	11647	11672	11678
11685	11691	11718	11743	11749	11756	11762	11773	11776	11783	11786	11964	11989
11995	12002	12008	12036	12040	12056	12073	12079	12090	12093	12100	12103	12281
12306	12313	12318	12326	12330	12336	12343	12390	12396	12407	12410	12417	12420
12598	12623	12629	12636	12643	12659	12694	12700	12707	12713	12724	12727	12734
12737	12915	12940	12946	12953	12959	12986	13011	13017	13024	13030	13041	13044
13051	13054	13232	13237	13257	13270	13276	13303	13328	13334	13341	13347	13358
13361	13368	13371	13374	13377	13380	13387	13593	13620	13645	13651	13658	13664
13675	13678	13685	13688	13733	13746	13749	13761	13754	13778	13781	13787	13799
13802	13816	13819	13826	13829	13884	13887	13994	14000	14007	14016	14019	14025
14041	14057	14067	14077	14087	14097	14107	14117	14129	14139	14149	14159	14169
14179	14189	14199	14215	14234	14247	14260	14273	14286	14299	14312	14325	14338
14365	14384	14387	14393	14396	14402	14405	14412	14415	14430	14452	14548	14560
14572	14578	14587	14593	14602	14617	14620	14636	14658	14661	14668	14671	14736
14742	14762	14765	14777	14780	14788	14849	14855	14858	14919	14925	14928	14983
14998	15032	15038	15045	15051								
1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#	1#
3548	2309	2389	2506	2586	2703	2783	2900	2980	3107	3196	3333	3431
4810	3628	3745	3826	3943	4022	4139	4219	4336	4416	4533	4613	4730
6132	4927	5007	5124	5194	5304	5389	5511	5596	5718	5803	5925	6010
7459	6217	6339	6424	6546	6631	6753	6838	6960	7045	7167	7252	7422
9187	7659	7899	7955	8000	8057	8088	8179	8336	8486	8576	8736	9110
10504	9236	9427	9498	9553	9744	9815	9870	10061	10132	10187	10378	10449
11963	10695	10766	10821	11012	11083	11138	11329	11400	11455	11646	11717	11772
13302	12034	12089	12280	12351	12406	12597	12668	12723	12914	12985	13040	13231
1462#	13357	13548	13619	13674	14006	14364	14401	14429	14616	14635		
1#	1568											
1#	1567											
1#	1741#											
1#	2283	2290	2306	2331	2337	2344	2350	2356	2370	2386	2411	2417
2424	2430	2436	2447	2480	2487	2503	2528	2534	2541	2547	2553	2567
2583	2608	2614	2621	2627	2633	2644	2677	2684	2700	2725	2731	2738
2744	2750	2764	2780	2805	2811	2818	2824	2830	2841	2874	2881	2897
2922	2928	2935	2941	2947	2961	2977	3002	3008	3015	3021	3027	3038

.BSMEM= 140000
 .BZ = 001400

.CNVRT 006160
 .CONVR 006154
 .CO = 000400
 .DATAC 007420
 .D&R = 000400

CROSS REFERENCE TABLE -- USER SYMBOLS

3072	3079	3098	3104	3129	3135	3145	3148	3154	3166	3181	3193	3218
3223	3231	3257	3261	3273	3279	3285	3289	3295	3309	3325	3331	3337
3342	3350	3376	3380	3392	3398	3404	3408	3414	3428	3444	3450	3456
3462	3470	3496	3500	3512	3518	3524	3528	3534	3548	3564	3570	3576
3582	3590	3616	3620	3632	3638	3644	3648	3654	3668	3684	3690	3696
3702	3710	3736	3740	3752	3758	3764	3768	3774	3788	3804	3810	3816
3822	3830	3856	3860	3872	3878	3884	3888	3894	3908	3924	3930	3936
3942	3950	3976	3980	3992	3998	4004	4008	4014	4028	4044	4050	4056
4062	4070	4096	4100	4112	4118	4124	4128	4134	4148	4164	4170	4176
4182	4190	4216	4220	4232	4238	4244	4248	4254	4268	4284	4290	4296
4302	4310	4336	4340	4352	4358	4364	4368	4374	4388	4404	4410	4416
4422	4430	4456	4460	4472	4478	4484	4488	4494	4508	4524	4530	4536
4542	4550	4576	4580	4592	4598	4604	4608	4614	4628	4644	4650	4656
4662	4670	4696	4700	4712	4718	4724	4728	4734	4748	4764	4770	4776
4782	4790	4816	4820	4832	4838	4844	4848	4854	4868	4884	4890	4896
4902	4910	4936	4940	4952	4958	4964	4968	4974	4988	5004	5010	5016
5022	5030	5056	5060	5072	5078	5084	5088	5094	5108	5124	5130	5136
5142	5150	5176	5180	5192	5198	5204	5208	5214	5228	5244	5250	5256
5262	5270	5296	5300	5312	5318	5324	5328	5334	5348	5364	5370	5376
5382	5390	5416	5420	5432	5438	5444	5448	5454	5468	5484	5490	5496
5502	5510	5536	5540	5552	5558	5564	5568	5574	5588	5604	5610	5616
5622	5630	5656	5660	5672	5678	5684	5688	5694	5708	5724	5730	5736
5742	5750	5776	5780	5792	5798	5804	5808	5814	5828	5844	5850	5856
5862	5870	5896	5900	5912	5918	5924	5928	5934	5948	5964	5970	5976
5982	5990	6016	6020	6032	6038	6044	6048	6054	6068	6084	6090	6096
6102	6110	6136	6140	6152	6158	6164	6168	6174	6188	6204	6210	6216
6222	6230	6256	6260	6272	6278	6284	6288	6294	6308	6324	6330	6336
6342	6350	6376	6380	6392	6398	6404	6408	6414	6428	6444	6450	6456
6462	6470	6496	6500	6512	6518	6524	6528	6534	6548	6564	6570	6576
6582	6590	6616	6620	6632	6638	6644	6648	6654	6668	6684	6690	6696
6702	6710	6736	6740	6752	6758	6764	6768	6774	6788	6804	6810	6816
6822	6830	6856	6860	6872	6878	6884	6888	6894	6908	6924	6930	6936
6942	6950	6976	6980	6992	6998	7004	7008	7014	7028	7044	7050	7056
7062	7070	7096	7100	7112	7118	7124	7128	7134	7148	7164	7170	7176
7182	7190	7216	7220	7232	7238	7244	7248	7254	7268	7284	7290	7296
7302	7310	7336	7340	7352	7358	7364	7368	7374	7388	7404	7410	7416
7422	7430	7456	7460	7472	7478	7484	7488	7494	7508	7524	7530	7536
7542	7550	7576	7580	7592	7598	7604	7608	7614	7628	7644	7650	7656
7662	7670	7696	7700	7712	7718	7724	7728	7734	7748	7764	7770	7776
7782	7790	7816	7820	7832	7838	7844	7848	7854	7868	7884	7890	7896
7902	7910	7936	7940	7952	7958	7964	7968	7974	7988	8004	8010	8016
8022	8030	8056	8060	8072	8078	8084	8088	8094	8108	8124	8130	8136
8142	8150	8176	8180	8192	8198	8204	8208	8214	8228	8244	8250	8256
8262	8270	8296	8300	8312	8318	8324	8328	8334	8348	8364	8370	8376
8382	8390	8416	8420	8432	8438	8444	8448	8454	8468	8484	8490	8496
8502	8510	8536	8540	8552	8558	8564	8568	8574	8588	8604	8610	8616
8622	8630	8656	8660	8672	8678	8684	8688	8694	8708	8724	8730	8736
8742	8750	8776	8780	8792	8798	8804	8808	8814	8828	8844	8850	8856
8862	8870	8896	8900	8912	8918	8924	8928	8934	8948	8964	8970	8976
8982	8990	9016	9020	9032	9038	9044	9048	9054	9068	9084	9090	9096
9102	9110	9136	9140	9152	9158	9164	9168	9174	9188	9204	9210	9216
9222	9230	9256	9260	9272	9278	9284	9288	9294	9308	9324	9330	9336
9342	9350	9376	9380	9392	9398	9404	9408	9414	9428	9444	9450	9456
9462	9470	9496	9500	9512	9518	9524	9528	9534	9548	9564	9570	9576
9582	9590	9616	9620	9632	9638	9644	9648	9654	9668	9684	9690	9696
9702	9710	9736	9740	9752	9758	9764	9768	9774	9788	9804	9810	9816
9822	9830	9856	9860	9872	9878	9884	9888	9894	9908	9924	9930	9936
9942	9950	9976	9980	9992	9998	10004	10008	10014	10028	10044	10050	10056
10062	10070	10096	10100	10112	10118	10124	10128	10134	10148	10164	10170	10176
10182	10190	10216	10220	10232	10238	10244	10248	10254	10268	10284	10290	10296
10302	10310	10336	10340	10352	10358	10364	10368	10374	10388	10404	10410	10416
10422	10430	10456	10460	10472	10478	10484	10488	10494	10508	10524	10530	10536
10542	10550	10576	10580	10592	10598	10604	10608	10614	10628	10644	10650	10656
10662	10670	10696	10700	10712	10718	10724	10728	10734	10748	10764	10770	10776
10782	10790	10816	10820	10832	10838	10844	10848	10854	10868	10884	10890	10896
10902	10910	10936	10940	10952	10958	10964	10968	10974	10988	11004	11010	11016
11022	11030	11056	11060	11072	11078	11084	11088	11094	11108	11124	11130	11136
11142	11150	11176	11180	11192	11198	11204	11208	11214	11228	11244	11250	11256
11262	11270	11296	11300	11312	11318	11324	11328	11334	11348	11364	11370	11376
11382	11390	11416	11420	11432	11438	11444	11448	11454	11468	11484	11490	11496
11502	11510	11536	11540	11552	11558	11564	11568	11574	11588	11604	11610	11616
11622	11630	11656	11660	11672	11678	11684	11688	11694	11708	11724	11730	11736
11742	11750	11776	11780	11792	11798	11804	11808	11814	11828	11844	11850	11856
11862	11870	11896	11900	11912	11918	11924	11928	11934	11948	11964	11970	11976
11982	11990	12016	12020	12032	12038	12044	12048	12054	12068	12084	12090	12096
12102	12110	12136	12140	12152	12158	12164	12168	12174	12188	12204	12210	12216
12222	12230	12256	12260	12272	12278	12284	12288	12294	12308	12324	12330	12336
12342	12350	12376	12380	12392	12398	12404	12408	12414	12428	12444	12450	12456
12462	12470	12496	12500	12512	12518	12524	12528	12534	12548	12564	12570	12576
12582	12590	12616	12620	12632	12638	12644	12648	12654	12668	12684	12690	12696
12702	12710	12736	12740	12752	12758	12764	12768	12774	12788	12804	12810	12816
12822	12830	12856	12860	12872	12878	12884	12888	12894	12908	12924	12930	12936
12942	12950	12976	12980	12992	12998	13004	13008	13014	13028	13044	13050	13056
13062	13070	13096	13100	13112	13118	13124	13128	13134	13148	13164	13170	13176
13182	13190	13216	13220	13232	13238	13244	13248	13254	13268	13284	13290	13296

B11

CROSS REFERENCE TABLE -- USER SYMBOLS

.DBRSH= 001400

.DBRSP= 003400

.DD = 003000

.DEC = 000160

.DELAY 007304

.DMEH = 002400

13364	13405	13520	13533	13557	13563	13570	13583	13601	13604	13628	13634	13641
13654	13666	13681	13714	13727	13743	13751	13757	13774	13783	13798	13812	13822
13857	13974	13980	14040	14053	14061	14067	14083	14063	14073	14083	14093	14103
14113	14125	14130	14144	14156	14164	14175	14186	14196	14204	14211	14217	14223
14230	14238	14247	14254	14260	14267	14273	14280	14288	14299	14305	14301	14308
14314	14321	14327	14334	14341	14348	14354	14360	14368	14376	14382	14435	14441
14448	14481	14511	14517	14524	14531	14537	14544	14550	14556	14568	14574	14589
14607	14613	14620	14627	14634	14641	14647	14654	14660	14666	14723	14732	14738
14747	14758	14764	14771	14778	14785	14791	14797	14804	14810	14894	14903	14909
14915	14921	14930	14936	14943	14949	14955	14961	14968	14974	15010	15016	15022
15028	15034	15041	15047	15054	15060	15067	15073	15080	15086			
1	2360	2740	2857	2837	2754	2834	2951	3031	3158	3247	3384	3482
3599	3679	3796	3876	3993	4073	4190	4270	4387	4467	4584	4664	4781
4861	4978	5058	5138	5255	5335	5450	5572	5657	5799	6064	6186	6271
6393	6478	6500	6585	6607	6682	7014	7099	7221	7306	7372	7564	7950
8230	8387											
11003	11071	9169	9418	9486	9735	9803	10052	10120	10369	10437	10686	10754
12973	13222	11320	11388	11637	11705	11954	12022	12271	12339	12588	12656	12905
2306	2386	2503	2573	2700	2780	2897	2977	3092	3104	3181	3193	3009
3324	3330	3407	3423	3438	3515	3625	3742	3822	3939	4019	4136	4216
4333	4413	4530	4610	4727	4807	4924	5004	5121	5191	5301	5386	5508
5593	5715	5800	5923	6007	6129	6214	6336	6421	6543	6628	6750	6835
6957	7042	7164	7249	7419	7486	7574	7611	7656	7696	7815	7861	8054
8176	8247	8333	8407	8483	8542	8673	8666	8702	8733	8792	9095	9096
9107	9166	9167	9178	9231	9412	9413	9418	9424	9483	9484	9486	9495
9548	9729	9730	9735	9741	9800	9801	9803	9812	9865	10046	10047	10052
10058	10117	10118	10120	10129	10182	10363	10364	10369	10375	10434	10435	10437
10446	10499	10680	10681	10686	10692	10751	10752	10754	10763	10816	10997	10998
11009	11068	11069	11080	11133	11134	11315	11320	11326	11365	11386	11388	11397
11450	11631	11632	11637	11643	11702	11703	11705	11714	11767	11948	11949	11954
11960	12019	12020	12022	12031	12084	12265	12266	12277	12336	12337	12348	12401
12582	12583	12594	12652	12654	12655	12718	12719	12790	12911	12970	12971	12982
13035	13216	13217	13222	13231	13284	13285	13290	13299	13352	13533	13534	13545
13604	13605	13616	13669	13680	13681	14328	14329	14541	14613	14632	14720	14729
14753	14827	14836	14842	14843	14906	14906	14912	14970	14976	14991		
1	7952	7997	8085	8901	8913	8982	9233	9299	9550	9616	9867	9933
10184	10250	10501	10567	10818	10884	11135	11201	11452	11518	11769	11835	12085
12152	12403	12469	12720	12786	13037	13103	13354	13420	13539	13607	13671	14003
14037	14398											
1562	1714											
1	2318	2324	2393	2404	2515	2521	2595	2601	2712	2718	2792	2798
2909	2915	2989	3052	3086	3116	3122	3175	3205	3211	3303	3318	3342
3348	3401	3416	3440	3446	3557	3572	3637	3643	3754	3760	3834	3840
3951	3957	4031	4037	4148	4154	4234	4234	4345	4351	4425	4431	4542
4548	4622	4628	4739	4745	4819	4825	4926	4942	5016	5022	5106	5121
5165	5176	5191	5235	5289	5325	5374	5380	5487	5502	5581	5587	5703
5709	5788	5794	5910	5916	5935	6001	6117	6123	6202	6208	6324	6330
6409	6415	6531	6537	6616	6622	6738	6744	6823	6829	6945	6951	7030
7036	7152	7158	7237	7243	7468	7671	7711	7809	7839	7946	7991	7994
8054	8069	8154	8164	8241	8247	8311	8321	8401	8407	8474	8690	8856
8875	8904	8988	8991	8994	8997	9000	9003	9006	9009	9012	9015	9018
9021	9024	9027	9030	9033	9036	9039	9042	9045	9048	9051	9054	9057
9060	9063	9066	9069	9072	9075	9078	9081	9085	9098	9305	9308	9311
9314	9317	9320	9323	9326	9329	9332	9335	9338	9341	9344	9347	9350
9353	9356	9359	9362	9365	9368	9371	9374	9377	9380	9383	9386	9389

D11

CROSS REFERENCE TABLE -- USER SYMBOLS

4231	4237	4339	4342	4348	4354	4419	4423	4428	4434	4536	4539	4545
4232	4516	4519	4522	4531	4533	4726	4729	4748	4813	4816	4823	4828
4233	4517	4539	4541	4540	4543	4730	4731	4749	4814	4818	4824	4829
4234	4518	4540	4542	4541	4544	4731	4732	4750	4815	4820	4826	4831
4235	4519	4541	4543	4542	4545	4732	4733	4751	4816	4821	4827	4832
4236	4520	4542	4544	4543	4546	4733	4734	4752	4817	4822	4828	4833
4237	4521	4543	4545	4544	4547	4734	4735	4753	4818	4823	4829	4834
4238	4522	4544	4546	4545	4548	4735	4736	4754	4819	4824	4830	4835
4239	4523	4545	4547	4546	4549	4736	4737	4755	4820	4825	4831	4836
4240	4524	4546	4548	4547	4550	4737	4738	4756	4821	4826	4832	4837
4241	4525	4547	4549	4548	4551	4738	4739	4757	4822	4827	4833	4838
4242	4526	4548	4550	4549	4552	4739	4740	4758	4823	4828	4834	4839
4243	4527	4549	4551	4550	4553	4740	4741	4759	4824	4829	4835	4840
4244	4528	4550	4552	4551	4554	4741	4742	4760	4825	4830	4836	4841
4245	4529	4551	4553	4552	4555	4742	4743	4761	4826	4831	4837	4842
4246	4530	4552	4554	4553	4556	4743	4744	4762	4827	4832	4838	4843
4247	4531	4553	4555	4554	4557	4744	4745	4763	4828	4833	4839	4844
4248	4532	4554	4556	4555	4558	4745	4746	4764	4829	4834	4840	4845
4249	4533	4555	4557	4556	4559	4746	4747	4765	4830	4835	4841	4846
4250	4534	4556	4558	4557	4560	4747	4748	4766	4831	4836	4842	4847
4251	4535	4557	4559	4558	4561	4748	4749	4767	4832	4837	4843	4848
4252	4536	4558	4560	4559	4562	4749	4750	4768	4833	4838	4844	4849
4253	4537	4559	4561	4560	4563	4750	4751	4769	4834	4839	4845	4850
4254	4538	4560	4562	4561	4564	4751	4752	4770	4835	4840	4846	4851
4255	4539	4561	4563	4562	4565	4752	4753	4771	4836	4841	4847	4852
4256	4540	4562	4564	4563	4566	4753	4754	4772	4837	4842	4848	4853
4257	4541	4563	4565	4564	4567	4754	4755	4773	4838	4843	4849	4854
4258	4542	4564	4566	4565	4568	4755	4756	4774	4839	4844	4850	4855
4259	4543	4565	4567	4566	4569	4756	4757	4775	4840	4845	4851	4856
4260	4544	4566	4568	4567	4570	4757	4758	4776	4841	4846	4852	4857
4261	4545	4567	4569	4568	4571	4758	4759	4777	4842	4847	4853	4858
4262	4546	4568	4570	4569	4572	4759	4760	4778	4843	4848	4854	4859
4263	4547	4569	4571	4570	4573	4760	4761	4779	4844	4849	4855	4860
4264	4548	4570	4572	4571	4574	4761	4762	4780	4845	4850	4856	4861
4265	4549	4571	4573	4572	4575	4762	4763	4781	4846	4851	4857	4862
4266	4550	4572	4574	4573	4576	4763	4764	4782	4847	4852	4858	4863
4267	4551	4573	4575	4574	4577	4764	4765	4783	4848	4853	4859	4864
4268	4552	4574	4576	4575	4578	4765	4766	4784	4849	4854	4860	4865
4269	4553	4575	4577	4576	4579	4766	4767	4785	4850	4855	4861	4866
4270	4554	4576	4578	4577	4580	4767	4768	4786	4851	4856	4862	4867
4271	4555	4577	4579	4578	4581	4768	4769	4787	4852	4857	4863	4868
4272	4556	4578	4580	4579	4582	4769	4770	4788	4853	4858	4864	4869
4273	4557	4579	4581	4580	4583	4770	4771	4789	4854	4859	4865	4870
4274	4558	4580	4582	4581	4584	4771	4772	4790	4855	4860	4866	4871
4275	4559	4581	4583	4582	4585	4772	4773	4791	4856	4861	4867	4872
4276	4560	4582	4584	4583	4586	4773	4774	4792	4857	4862	4868	4873
4277	4561	4583	4585	4584	4587	4774	4775	4793	4858	4863	4869	4874
4278	4562	4584	4586	4585	4588	4775	4776	4794	4859	4864	4870	4875
4279	4563	4585	4587	4586	4589	4776	4777	4795	4860	4865	4871	4876
4280	4564	4586	4588	4587	4590	4777	4778	4796	4861	4866	4872	4877
4281	4565	4587	4589	4588	4591	4778	4779	4797	4862	4867	4873	4878
4282	4566	4588	4590	4589	4592	4779	4780	4798	4863	4868	4874	4879
4283	4567	4589	4591	4590	4593	4780	4781	4799	4864	4869	4875	4880
4284	4568	4590	4592	4591	4594	4781	4782	4800	4865	4870	4876	4881
4285	4569	4591	4593	4592	4595	4782	4783	4801	4866	4871	4877	4882
4286	4570	4592	4594	4593	4596	4783	4784	4802	4867	4872	4878	4883
4287	4571	4593	4595	4594	4597	4784	4785	4803	4868	4873	4879	4884
4288	4572	4594	4596	4595	4598	4785	4786	4804	4869	4874	4880	4885
4289	4573	4595	4597	4596	4599	4786	4787	4805	4870	4875	4881	4886
4290	4574	4596	4598	4597	4600	4787	4788	4806	4871	4876	4882	4887
4291	4575	4597	4599	4598	4601	4788	4789	4807	4872	4877	4883	4888
4292	4576	4598	4600	4599	4602	4789	4790	4808	4873	4878	4884	4889
4293	4577	4599	4601	4600	4603	4790	4791	4809	4874	4879	4885	4890
4294	4578	4600	4602	4601	4604	4791	4792	4810	4875	4880	4886	4891
4295	4579	4601	4603	4602	4605	4792	4793	4811	4876	4881	4887	4892
4296	4580	4602	4604	4603	4606	4793	4794	4812	4877	4882	4888	4893
4297	4581	4603	4605	4604	4607	4794	4795	4813	4878	4883	4889	4894
4298	4582	4604	4606	4605	4608	4795	4796	4814	4879	4884	4890	4895
4299	4583	4605	4607	4606	4609	4796	4797	4815	4880	4885	4891	4896
4300	4584	4606	4608	4607	4610	4797	4798	4816	4881	4886	4892	4897
4301	4585	4607	4609	4608	4611	4798	4799	4817	4882	4887	4893	4898
4302	4586	4608	4610	4609	4612	4799	4800	4818	4883	4888	4894	4899
4303	4587	4609	4611	4610	4613	4800	4801	4819	4884	4889	4895	4900
4304	4588	4610	4612	4611	4614	4801	4802	4820	4885	4890	4896	4901
4305	4589	4611	4613	4612	4615	4802	4803	4821	4886	4891	4897	4902
4306	4590	4612	4614	4613	4616	4803	4804	4822	4887	4892	4898	4903
4307	4591	4613	4615	4614	4617	4804	4805	4823	4888	4893	4899	4904
4308	4592	4614	4616	4615	4618	4805	4806	4824	4889	4894	4900	4905
4309	4593	4615	4617	4616	4619	4806	4807	4825	4890	4895	4901	4906
4310	4594	4616	4618	4617	4620	4807	4808	4826	4891	4896	4902	4907
4311	4595	4617	4619	4618	4621	4808	4809	4827	4892	4897	4903	4908
4312	4596	4618	4620	4619	4622	4809	4810	4828	4893	4898	4904	4909
4313	4597	4619	4621	4620	4623	4810	4811	4829	4894	4899	4905	4910
4314	4598	4620	4622	4621	4624	4811	4812	4830	4895	4900	4906	4911
4315	4599	4621	4623	4622	4625	4812	4813	4831	4896	4901	4907	4912
4316	4600	4622	4624	4623	4626	4813	4814	4832	4897	4902	4908	4913
4317	4601	4623	4625	4624	4627	4814	4815	4833	4898	4903	4909	4914
4318	4602	4624	4626	4625	4628	4815	4816	4834	4899	4904	4910	4915
4319	4603	4625	4627	4626	4629	4816	4817	4835	4900	4905	4911	4916
4320	4604	4626	4628	4627	4630	4817	4818	4836	4901	4906	4912	4917
4321	4605	4627	4629	4628	4631	4818	4819	4837	4902	4907	4913	4918
4322	4606	4628	4630	4629	4632	4819	4820	4838	4903	4908	4914	4919
4323	4607	4629	4631	4630	4633	4820	4821	4839	4904	4909	4915	4920
4324	4608	4630	4632	4631	4634	4821	4822	4840	4905	4910	4916	4921
4325	4609	4631	4633	4632	4635	4822	4823	4841	4906	4911	4917	4922
4326	4610	4632	4634	4633	4636	4823	4824	4842	4907	4912	4918	4923
4327	4611	4633	4635	4634	4637	4824	4825	4843	4908	4913	4919	4924
4328	4612	4634	4636	4635	4638	4825	4826	4844	4909	4914	4920	4925
4329	4613	4635	4637	4636	4639	4826	4827	4845	4910	4915	4921	4926
4330	4614	4636	4638	4637	4640							

CROSS REFERENCE TABLE -- USER SYMBOLS

.DO = 000400
.FO = 000020

.INC = 000060

.LORN = 000240
.MINUS = 000360
.MSTCL 007334
.MO = 004000
.OR = 000300

.PLUS = 000000
.RESOS 006122
.ROMCL 007352
.SAVOS 006062
.SBR = 060000

12976	13036	13038	13091	13094	13097	13100	13104	13106	13205	13219	13222	13281
13290	13323	13323	13355	13408	13411	13414	13417	13421	13423	13522	13536	13599
13610	13669	13672	13717	13720	13723	13726	13790	13793	13860	13863	13866	13927
13930	13933	13936	13939	13942	13945	13948	13951	13954	13957	13959	13962	13971
14004	14009	14013	14031	14034	14038	14046	14049	14059	14069	14079	14089	14099
14109	14121	14131	14141	14151	14161	14171	14181	14191	14207	14220	14226	14239
14252	14261	14278	14291	14304	14317	14330	14349	14353	14390	14399	14417	14508
14532	14536	14541	14580	14583	14595	14598	14604	14623	14702	14705	14708	14711
14717	14720	14726	14729	14750	14753	14768	14771	14774	14810	14813	14818	14821
14827	14830	14836	14842	14863	14869	14886	14889	14891	14900	14906	14912	14933
14939	14955	14964	14970	14976	14988	14991						
1	1	1	1	1	1	1	1	1	1	1	1	1
2688	2697	2715	2777	2795	2494	2512	2574	2592	2691	2709	2771	2789
2906	2906	2928	2936	3098	3113	3187	3202	3315	3339	3413	3437	3536
3554	3616	3634	3733	3751	3813	3831	3930	3948	4010	4028	4127	4145
4207	4225	4324	4342	4404	4422	4521	4539	4601	4619	4718	4736	4798
4816	4915	4933	4995	5013	7407	7413	7450	7551	7608	7614	7641	7647
7650	7687	7690	7812	7824	7858	7985	7991	7994	13729	13757	13783	14049
14059	14069	14079	14089	14099	14109	14121	14131	14141	14151	14161	14171	14181
14191	14572	14544	14556	14568	14574	14580	14583	14589	14595	14598	14604	14623
14717	14726	14732	14738	14750	14758	14784	14824	14830	14845	14851	14900	14915
14921	14964	14979	14988	14994	15028	15034	15047					
1	1	1	1	1	1	1	1	1	1	1	1	1
8910	7830	8003	8006	8091	8094	8235	8395	8530	8614	8693	8780	8862
1	1	1	1	1	1	1	1	1	1	1	1	1
1561	1725											
1	1	1	1	1	1	1	1	1	1	1	1	1
14842	14912	14976	7554	7611	7815	10052	10120	14541	14720	14729	14753	14827
1	1	1	1	1	1	1	1	1	1	1	1	1
1450	1560											
1563	1730											
1436	1559											
1	1	1	1	1	1	1	1	1	1	1	1	1
2392	2286	2294	2300	2306	2312	2337	2350	2356	2360	2374	2380	2386
2557	2417	2430	2436	2440	2463	2491	2497	2503	2509	2534	2547	2553
2706	2571	2577	2583	2589	2614	2627	2633	2637	2680	2688	2694	2700
2877	2731	2744	2750	2754	2768	2774	2780	2786	2811	2824	2830	2834
3008	2885	2891	2897	2903	2928	2941	2947	2951	2965	2971	2977	2983
3158	3021	3027	3031	3075	3083	3092	3095	3104	3110	3135	3148	3154
3312	3172	3181	3184	3193	3199	3224	3237	3243	3247	3292	3300	3309
3434	3324	3330	3336	3361	3374	3380	3384	3398	3407	3410	3422	3428
3599	3434	3472	3478	3482	3525	3533	3539	3545	3551	3576	3589	3595
3748	3613	3619	3625	3631	3656	3669	3675	3679	3722	3730	3736	3742
3919	3773	3786	3792	3796	3810	3816	3822	3828	3853	3866	3872	3876
4050	3927	3933	3939	3945	3970	3983	3989	3993	4007	4013	4019	4025
4204	4063	4069	4073	4116	4124	4130	4136	4142	4167	4180	4186	4190
4364	4210	4216	4222	4247	4260	4266	4270	4313	4321	4327	4333	4339
4518	4377	4383	4387	4401	4407	4413	4419	4444	4457	4463	4467	4510
4654	4524	4530	4536	4561	4574	4580	4584	4598	4604	4610	4616	4641
4801	4660	4664	4707	4715	4721	4727	4733	4758	4771	4777	4781	4795
4968	4807	4813	4838	4851	4857	4861	4904	4912	4918	4924	4930	4955
5130	4974	4978	4992	4998	5004	5010	5035	5048	5054	5058	5102	5115
5301	5159	5162	5165	5185	5200	5229	5232	5235	5279	5286	5289	5295
5398	5307	5310	5313	5319	5325	5358	5371	5374	5380	5386	5392	5395
	5404	5410	5443	5486	5453	5496	5502	5508	5514	5517	5520	5526

CROSS REFERENCE TABLE -- USER SYMBOLS

5532	5565	5578	5581	5587	5593	5599	5602	5605	5611	5617	5650	5693
5700	5703	5709	5715	5721	5724	5727	5733	5739	5772	5786	5788	5794
5800	5806	5809	5812	5818	5824	5857	5800	5907	5910	5916	5922	5928
5931	5934	5940	5946	5979	5982	5995	6001	6007	6013	6016	6019	6025
6031	6034	6107	6114	6117	6123	6129	6135	6138	6141	6147	6153	6186
6199	6202	6208	6214	6220	6223	6226	6232	6238	6271	6314	6321	6324
6330	6336	6343	6349	6348	6351	6350	6393	6406	6409	6415	6421	6427
6430	6436	6439	6444	6478	6471	6478	6531	6537	6543	6549	6552	6555
6561	6567	6568	6574	6616	6613	6616	6634	6637	6640	6646	6652	6685
6728	6735	6738	6744	6750	6753	6759	6782	6768	6774	6807	6820	6823
6829	6835	6838	6844	6847	6853	6859	6892	6935	6942	6945	6951	6957
6963	6966	6969	6975	6981	7014	7027	7030	7036	7042	7048	7051	7054
7060	7066	7099	7143	7149	7152	7158	7164	7170	7173	7176	7182	7188
7221	7234	7237	7243	7249	7255	7258	7261	7267	7273	7306	7347	7350
7353	7356	7359	7363	7368	7372	7374	7380	7386	7389	7392	7395	7398
7404	7419	7415	7428	7431	7437	7456	7462	7465	7521	7524	7527	7530
7533	7536	7539	7545	7545	7548	7554	7560	7564	7566	7572	7578	7581
7584	7587	7590	7593	7598	7599	7602	7611	7623	7626	7629	7632	7638
7656	7663	7665	7668	7686	7702	7705	7708	7764	7767	7770	7773	7776
7779	7782	7785	7788	7791	7794	7800	7806	7821	7830	7861	7907	7910
7913	7916	7919	7925	7928	7934	7946	7950	7952	7967	7970	7973	7976
7982	7997	8000	8003	8009	8014	8024	8027	8030	8045	8048	8063	8066
8085	8091	8093	8098	8103	8109	8117	8121	8127	8191	8194	8197	8207
8220	8226	8231	8237	8243	8249	8257	8261	8267	8327	8333	8339	8348
8351	8357	8364	8370	8376	8382	8390	8393	8414	8417	8459	8468	8471
8474	8480	8483	8489	8495	8501	8507	8510	8533	8542	8561	8564	8573
8582	8588	8591	8597	8617	8623	8630	8636	8681	8690	8693	8696	8702
8721	8727	8733	8740	8745	8751	8761	8774	8780	8783	8792	8838	8847
8850	8857	8863	8870	8875	8881	8901	8913	8916	8919	8970	8973	8976
8979	8985	8991	8997	9003	9009	9010	9122	9128	9151	9157	9166	9169
9187	9193	9199	9205	9211	9217	9223	9233	9237	9243	9296	9299	9302
9412	9418	9424	9430	9436	9442	9474	9483	9504	9510	9516	9526	9539
9548	9550	9556	9562	9568	9574	9610	9616	9629	9640	9646	9662	9672
9785	9791	9800	9806	9812	9818	9827	9843	9855	9867	9870	9884	9894
9930	9933	9936	10173	10182	10189	10194	10199	10205	10210	10217	10224	10239
10150	10160	10170	10173	10182	10189	10194	10199	10205	10210	10217	10224	10239
10390	10396	10406	10406	10419	10425	10434	10435	10441	10455	10467	10477	10490
10555	10558	10561	10561	10564	10567	10570	10570	10580	10585	10597	10613	10619
10751	10772	10778	10778	10784	10794	10807	10816	10818	10822	10872	10875	10881
10887	10997	11003	11003	11018	11024	11030	11040	11040	11053	11059	11078	11081
11101	11111	11124	11124	11133	11135	11139	11149	11149	11155	11198	11201	11209
11341	11347	11357	11357	11370	11376	11385	11406	11412	11418	11428	11441	11450
11506	11509	11512	11512	11515	11518	11521	11631	11652	11658	11664	11674	11687
11702	11723	11729	11729	11735	11745	11758	11767	11769	11823	11826	11829	11832
11838	11948	11969	11969	11975	11981	11991	12004	12010	12019	12040	12046	12052
12075	12084	12086	12140	12140	12143	12146	12149	12152	12155	12265	12271	12286
12298	12308	12321	12327	12336	12336	12339	12357	12357	12369	12379	12392	12401
12457	12460	12463	12466	12469	12469	12472	12582	12582	12589	12599	12615	12625
12644	12653	12656	12674	12680	12680	12686	12696	12709	12718	12720	12774	12780
12783	12786	12789	12899	12905	12905	12920	12926	12932	12942	12951	12970	12973
12991	12997	13003	13013	13026	13026	13035	13037	13091	13094	13097	13100	13106
13216	13237	13243	13249	13259	13259	13272	13278	13287	13308	13314	13320	13343
13352	13354	13408	13411	13414	13414	13417	13420	13423	13533	13539	13554	13566
13576	13589	13595	13604	13607	13607	13625	13631	13637	13647	13660	13669	13717
13720	13723	13726	13754	13789	13789	13792	13795	13860	13863	13866	13876	13882
13885	13888	13892	13895	13900	13900	13903	13906	13909	13912	13915	13918	13927

CROSS REFERENCE TABLE -- USER SYMBOLS

13930	13933	13936	13939	13942	13945	13948	13951	13954	13957	13962	13965	13968
13971	13980	13990	13996	14003	14009	14012	14037	14046	14204	14207	14217	14220
14223	14226	14236	14239	14249	14252	14252	14265	14275	14278	14288	14291	14301
14304	14314	14317	14327	14330	14349	14361	14370	14376	14389	14398	14417	14420
14426	14432	14435	14438	14444	14490	14493	14496	14499	14502	14505	14508	14514
14520	14526	14541	14553	14565	14607	14613	14626	14632	14638	14641	14644	14650
14702	14705	14708	14711	14720	14729	14753	14767	14770	14773	14809	14812	14815
14818	14827	14836	14842	14860	14863	14869	14872	14885	14888	14891	14897	14906
14912	14930	14933	14939	14942	14955	14961	14970	14976	14991	15013	15019	15025
10868	1558											
2731	2744	2750	2811	2824	2830	2828	2941	2947	3008	3021	3027	3135
3148	3154	3224	3237	3243	3361	3374	3380	3459	3472	3478	3576	3589
3595	3656	3669	3675	3773	3786	3792	3853	3866	3872	3970	3983	3989
4050	4062	4069	4167	4180	4186	4247	4260	4266	4364	4377	4383	4444
4457	4462	4561	4574	4580	4641	4654	4660	4758	4771	4777	4838	4851
4857	4955	4968	4974	5038	5048	5054	5165	5235	5289	5310	5374	5395
5496	5517	5581	5602	5703	5724	5788	5809	5910	5931	5995	6016	6117
6138	6202	6223	6242	6343	6409	6430	6531	6552	6616	6637	6738	6759
6823	6844	6945	6986	7030	7051	7152	7173	7237	7258	7389	7395	7428
7465	7581	7587	7593	7599	7634	7652	7665	7705	7967	7973	8045	8048
8063	8194	8207	8220	8226	8254	8257	8351	8364	8377	8383	8414	8417
8474	8492	8504	8582	8590	8742	8761	8774	8868	8919	9101	9138	9151
9157	9169	9209	9230	9330	9468	9474	9526	9539	9772	9785	9791	9843
9856	10089	10102	10108	10160	10173	10406	10419	10425	10477	10490	10723	10736
10742	10794	10907	11040	11053	11059	11111	11124	11357	11370	11376	11428	11441
11674	11687	11693	11745	11758	11911	12004	12010	12062	12075	12308	12321	12327
12379	12392	12623	12638	12644	12696	12709	12942	12955	12961	13013	13026	13259
13272	13278	13330	13343	13357	13399	13505	13647	13660	13795	13990	13996	14204
14217	14223	14236	14249	14255	14275	14285	14301	14314	14327	14420	14435	14456
14607	14626	14641	14660	14676	14690	14697	15001					
2440	2286	2294	2300	2311	2319	2327	2374	2380	2392	2401	2407	
2637	2483	2491	2500	2511	2519	2527	2571	2577	2589	2598	2604	
2834	2680	2688	2699	2706	2715	2721	2768	2774	2786	2788	2801	
3031	3075	3083	3089	3096	3102	3118	3158	3171	3183	3184	3198	
3208	3214	3247	3250	3256	3260	3272	3308	3317	3318	3319	3349	
3404	3410	3419	3420	3423	3424	3425	3458	3459	3459	3464	3498	
3599	3613	3619	3621	3624	3625	3626	3633	3633	3633	3660	3666	
3796	3810	3816	3821	3827	3837	3846	3876	3877	3877	3757	3763	
3993	4007	4013	4018	4024	4034	4040	4073	4073	4073	3945	3960	
4190	4204	4210	4220	4221	4221	4227	4116	4124	4130	4142	4157	
4387	4401	4407	4419	4428	4434	4437	4313	4321	4327	4339	4354	
4584	4598	4604	4616	4621	4621	4627	4510	4518	4524	4536	4551	
4781	4795	4801	4813	4821	4821	4827	4667	4674	4674	4686	4748	
4978	4992	4998	5010	5019	5021	5027	4707	4715	4721	4733	4748	
5156	5179	5182	5185	5197	5205	5205	4904	4912	4918	4930	4945	
5319	5325	5335	5348	5354	5358	5358	5102	5109	5112	5115	5130	
5420	5433	5439	5443	5486	5498	5498	5277	5286	5290	5298	5313	
5555	5561	5565	5578	5584	5587	5599	5371	5380	5380	5388	5410	
5650	5693	5700	5706	5709	5721	5727	5499	5514	5517	5527	5542	
5785	5791	5794	5806	5812	5818	5824	5605	5611	5617	5627	5646	
5913	5916	5928	5934	5940	5946	5956	5733	5739	5749	5762	5772	
6013	6019	6025	6031	6041	6054	6060	5834	5847	5853	5857	5907	
6141	6147	6153	6163	6176	6182	6186	5969	5975	5979	5988	6001	
6238	6248	6261	6267	6271	6314	6321	6064	6107	6114	6120	6135	
							6199	6205	6208	6220	6232	
							6327	6330	6342	6348	6360	

.SCOP1 004360
.SELA = 000200

.SELB = 000220

CROSS REFERENCE TABLE -- USER SYMBOLS

6370	6383	6389	6393	6406	6412	6415	6427	6433	6439	6445	6455	6468
6474	6478	6521	6528	6534	6537	6549	6555	6561	6567	6577	6590	6596
6600	6613	6619	6622	6634	6640	6646	6652	6662	6675	6681	6685	6728
6735	6741	6744	6755	6762	6768	6774	6784	6797	6803	6807	6820	6826
6829	6841	6847	6853	6859	6869	6882	6888	6892	6935	6942	6948	6951
6963	6969	6975	6981	6991	7004	7010	7014	7027	7033	7036	7048	7054
7060	7066	7076	7089	7095	7099	7142	7149	7155	7158	7170	7176	7182
7188	7198	7211	7217	7221	7234	7240	7243	7255	7261	7267	7273	7283
7296	7302	7306	7317	7350	7353	7356	7359	7362	7368	7372	7374	7380
7386	7395	7398	7404	7425	7431	7437	7462	7471	7521	7524	7527	7530
7533	7536	7539	7542	7545	7548	7560	7564	7566	7572	7578	7584	7590
7596	7602	7626	7632	7638	7662	7668	7674	7702	7708	7714	7764	7767
7770	7773	7776	7779	7782	7785	7788	7791	7794	7800	7806	7821	7842
7907	7910	7913	7916	7919	7922	7928	7934	7946	7950	7970	7976	7982
8021	8024	8027	8030	8042	8050	8056	8072	8142	8157	8160	8164	8167
8170	8182	8185	8191	8197	8230	8299	8314	8317	8321	8324	8327	8339
8342	8348	8354	8387	8459	8468	8471	8477	8489	8495	8501	8507	8561
8564	8567	8579	8585	8591	8672	8681	8684	8721	8724	8727	8739	8745
8751	8838	8847	8850	8853	8859	8878	8907	8970	8973	8976	8979	8985
9098	9113	9116	9122	9128	9160	9172	9184	9187	9193	9199	9287	9290
9293	9296	9302	9415	9430	9433	9439	9445	9477	9489	9501	9504	9510
9516	9604	9607	9610	9613	9619	9732	9747	9750	9756	9762	9794	9806
9818	9821	9827	9833	9921	9924	9927	9930	9936	10049	10064	10067	10073
10079	10111	10123	10135	10138	10144	10150	10238	10241	10244	10247	10253	10366
10381	10384	10390	10396	10428	10440	10452	10455	10461	10467	10555	10558	10561
10564	10570	10683	10698	10701	10707	10713	10745	10757	10769	10772	10778	10784
10872	10875	10878	10881	10887	11000	11015	11018	11024	11030	11062	11074	11086
11089	11095	11101	11189	11192	11195	11198	11204	11317	11332	11335	11341	11347
11379	11391	11403	11406	11412	11418	11505	11509	11512	11515	11521	11634	11649
11652	11658	11664	11696	11708	11720	11723	11729	11735	11823	11826	11829	11832
11838	11951	11966	11969	11975	11981	12013	12025	12037	12040	12046	12052	12140
12143	12146	12149	12155	12228	12233	12236	12242	12272	12330	12342	12354	12357
12363	12369	12457	12460	12466	12472	12477	12525	12500	12603	12609	12615	12647
12659	12671	12674	12680	12686	12774	12777	12780	12783	12789	12902	12917	12920
12926	12932	12964	12976	12988	12991	12997	13003	13003	13091	13094	13097	13106
13219	13234	13237	13243	13249	13281	13293	13305	13305	13308	13314	13320	13411
13414	13417	13423	13526	13551	13554	13560	13566	13598	13610	13622	13625	13631
13637	13717	13720	13723	13726	13738	13754	13770	13808	13610	13663	13666	13673
13879	13882	13885	13888	13892	13895	13900	13903	13906	13909	13912	13915	13918
13921	13927	13930	13933	13936	13939	13942	13945	13948	13951	13954	13957	13962
13971	14009	14016	14207	14220	14226	14239	14252	14265	14278	14291	14304	14317
14330	14349	14356	14367	14370	14376	14417	14432	14438	14444	14490	14493	14496
14499	14502	14505	14508	14514	14520	14526	14553	14565	14638	14644	14650	14702
14705	14708	14711	14815	14818	14869	14872	14891	14897	14939	14942	14955	14961
15013	15019	15025	2318	2324	2331	2344	2370	2398	2404	2411	2424	2447
2480	2487	2515	2521	2528	2541	2567	2595	2601	2608	2621	2644	2677
2684	2712	2718	2725	2738	2764	2792	2798	2805	2818	2841	2874	2881
2909	2915	2922	2935	2961	2989	2995	3002	3015	3038	3072	3079	3086
3116	3122	3129	3142	3158	3175	3205	3211	3218	3231	3254	3289	3296
3303	3318	3342	3348	3355	3368	3394	3401	3416	3440	3446	3452	3466
3489	3522	3529	3557	3563	3570	3583	3609	3637	3643	3650	3662	3686
3719	3726	3754	3760	3767	3780	3806	3834	3840	3847	3860	3883	3916
3923	3951	3957	3964	3977	4003	4031	4037	4044	4057	4080	4113	4120
4148	4154	4161	4174	4200	4228	4234	4241	4254	4277	4310	4317	4345
4351	4358	4371	4397	4425	4431	4438	4451	4474	4507	4514	4542	4548

.SIMM = 000000

CROSS REFERENCE TABLE -- USER SYMBOLS

4557	4568	4594	4623	4638	4648	4671	4704	4711	4739	4745	4752
4795	4791	4819	4823	4828	4848	4901	4908	4936	4942	4949	4962
4988	5016	5022	5029	5030	5039	5106	5133	5137	5147	5176	5203
5257	5279	5282	5286	5287	5288	5288	5288	5288	5288	5288	5288
5657	5690	5707	5730	5736	5743	5782	5815	5821	5828	5841	5864
5897	5904	5937	5943	5946	5952	6022	6028	6035	6048	6071	6104
6111	6144	6150	6157	6170	6186	6239	6242	6255	6278	6311	6318
6351	6357	6364	6377	6403	6436	6442	6449	6485	6518	6524	6558
6564	6571	6584	6610	6643	6649	6669	6692	6725	6732	6765	6771
6778	6791	6817	6850	6856	6863	6876	6899	6939	6972	6978	6985
6998	7024	7057	7063	7070	7083	7106	7139	7146	7185	7192	7205
7231	7264	7270	7277	7290	7313	7344	7365	7377	7383	7401	7441
7468	7475	7485	7518	7557	7569	7575	7605	7635	7671	7678	7718
7728	7761	7797	7803	7809	7818	7839	7846	7855	7865	7898	7904
7925	7931	7937	7940	7943	7961	7964	7979	8015	8018	8033	8039
8059	8076	8104	8139	8145	8148	8151	8154	8188	8201	8214	8264
8228	8302	8305	8308	8311	8324	8328	8371	8401	8424	8455	8465
8498	8511	8521	8536	8552	8555	8558	8588	8595	8605	8620	8669
8675	8678	8687	8712	8715	8718	8748	8755	8769	8783	8803	8841
8744	8856	8875	8882	8904	8908	8926	8931	8949	8967	8988	8994
8997	9000	9003	9006	9009	9012	9015	9018	9021	9024	9027	9033
9036	9039	9042	9045	9048	9051	9054	9057	9060	9063	9066	9072
9075	9078	9081	9084	9088	9092	9119	9125	9132	9145	9166	9190
9203	9216	9228	9243	9278	9281	9284	9305	9308	9311	9314	9317
9323	9326	9329	9332	9335	9338	9341	9344	9347	9350	9353	9356
9362	9365	9368	9371	9374	9377	9380	9383	9386	9389	9392	9395
9401	9405	9409	9436	9442	9445	9462	9480	9507	9513	9520	9533
9560	9595	9598	9601	9622	9625	9631	9634	9637	9637	9640	9643
9649	9652	9655	9658	9661	9664	9667	9670	9673	9676	9679	9682
9688	9691	9694	9697	9700	9703	9706	9709	9712	9715	9718	9722
9753	9759	9766	9779	9797	9804	9830	9837	9850	9862	9877	9912
9918	9939	9942	9945	9948	9951	9954	9957	9960	9963	9966	9969
9975	9978	9981	9984	9987	9990	9993	9996	9999	10002	10005	10008
10014	10017	10020	10023	10026	10029	10032	10035	10039	10043	10047	10051
10096	10114	10141	10147	10154	10167	10176	10194	10229	10232	10236	10239
10262	10265	10268	10271	10274	10277	10280	10283	10286	10289	10292	10295
10301	10304	10307	10310	10313	10316	10319	10322	10325	10328	10331	10334
10340	10343	10346	10349	10352	10355	10358	10361	10364	10367	10370	10373
10464	10471	10484	10496	10511	10546	10549	10552	10573	10400	10413	10431
10588	10591	10594	10597	10600	10603	10606	10609	10612	10576	10579	10582
10627	10630	10633	10636	10639	10642	10645	10648	10651	10615	10618	10621
10666	10669	10673	10676	10679	10682	10685	10688	10691	10654	10657	10660
10813	10828	10863	10866	10869	10872	10875	10878	10881	10725	10781	10788
10914	10917	10920	10923	10926	10929	10932	10935	10938	10902	10905	10908
10953	10956	10959	10962	10965	10968	10971	10974	10977	10941	10944	10947
10994	11021	11027	11034	11047	11065	11092	11098	11098	10960	10963	10966
11183	11186	11207	11210	11213	11216	11219	11222	11225	11118	11130	11145
11240	11243	11246	11249	11252	11255	11258	11261	11264	11228	11231	11234
11279	11282	11285	11288	11291	11294	11297	11300	11303	11267	11270	11273
11351	11364	11382	11409	11415	11422	11435	11447	11462	11307	11311	11338
11527	11530	11533	11536	11539	11542	11545	11548	11551	11497	11500	11503
11566	11569	11572	11575	11578	11581	11584	11587	11590	11554	11557	11560
11605	11608	11611	11614	11617	11620	11624	11628	11655	11593	11596	11599
11726	11732	11739	11752	11764	11779	11814	11817	11820	11661	11668	11681
11853	11856	11859	11862	11865	11868	11871	11874	11877	11841	11844	11847

CROSS REFERENCE TABLE -- USER SYMBOLS

11892	11895	11898	11901	11904	11907	11910	11913	11916	11919	11922	11925	11928
11931	11934	11937	11941	11945	11972	11978	11995	11998	12016	12043	12049	12056
12069	12081	12096	12131	12134	12137	12158	12161	12164	12167	12170	12173	12176
12179	12182	12185	12188	12191	12194	12197	12200	12203	12206	12209	12212	12215
12218	12221	12224	12227	12230	12233	12236	12239	12242	12245	12248	12251	12254
12258	12261	12264	12267	12270	12273	12276	12279	12282	12285	12288	12291	12294
12448	12451	12454	12457	12478	12481	12484	12487	12490	12493	12496	12499	12502
12505	12508	12511	12514	12517	12520	12523	12526	12529	12532	12535	12538	12541
12544	12547	12550	12553	12556	12559	12562	12565	12568	12571	12574	12577	12580
12612	12619	12632	12650	12677	12683	12690	12703	12715	12730	12765	12768	12771
12792	12795	12798	12801	12804	12807	12810	12813	12816	12819	12822	12825	12828
12831	12834	12837	12840	12843	12846	12849	12852	12855	12858	12861	12864	12867
12870	12873	12876	12879	12882	12885	12888	12891	12894	12897	12900	12903	12906
12967	12994	13000	13007	13020	13032	13047	13062	13085	13088	13099	13109	13112
13118	13121	13124	13127	13130	13133	13136	13139	13142	13145	13148	13151	13154
13157	13160	13163	13166	13169	13172	13175	13178	13181	13184	13187	13190	13193
13196	13199	13202	13205	13209	13213	13240	13246	13253	13266	13284	13311	13317
13324	13337	13349	13364	13399	13402	13405	13426	13429	13432	13435	13438	13441
13444	13447	13450	13453	13456	13459	13462	13465	13468	13471	13474	13477	13480
13483	13486	13489	13492	13495	13498	13501	13504	13507	13510	13513	13516	13519
13522	13526	13530	13557	13563	13570	13583	13601	13628	13634	13641	13654	13666
13681	13714	13735	13742	13751	13767	13774	13805	13812	13822	13857	13870	13873
13974	14021	14027	14031	14034	14053	14063	14073	14093	14093	14103	14113	14125
14135	14145	14155	14165	14175	14185	14185	14211	14220	14220	14256	14269	14282
14295	14308	14321	14334	14343	14346	14352	14373	14380	14380	14441	14448	14481
14484	14487	14511	14517	14523	14529	14538	14550	14562	14562	14654	14664	14699
14714	14723	14747	14821	14823	14839	14866	14894	14903	14903	14936	14958	14967
14973	14985	15010	15016	15022	15041	15041	15041	15041	15041	15041	15041	15041
4127	3536	3554	3616	3634	3733	3751	3813	3831	3831	3948	4010	4028
4736	4145	4207	4225	4324	4343	4404	4423	4521	4539	4601	4619	4718
7991	4798	4816	4915	4933	4995	5013	7413	7450	7647	7650	7687	7690
2888	7994	14121	14131	14141	14151	14161	14171	14181	14191	14604	14623	2789
7551	2297	2315	2377	2395	2494	2512	2574	2592	2691	2709	2771	7407
14069	2906	2968	2986	3098	3113	3187	3202	3315	3339	3413	3437	14059
14595	7608	7614	7641	7812	7824	7858	7885	13729	13757	13783	14049	14589
14900	14079	14089	14099	14109	14532	14544	14556	14568	14574	14580	14583	14651
2912	14598	14717	14726	14732	14738	14750	14758	14784	14824	14830	14845	2801
3351	14915	14921	14964	14979	14988	14994	15028	15034	15047	15047	2721	3345
3954	2321	2327	2401	2407	2518	2524	2598	2604	2715	2721	2795	3843
4551	2918	2992	2998	3089	3119	3125	3178	3208	3214	3306	3321	3945
5121	3404	3419	3443	3449	3560	3566	3640	3646	3757	3763	3837	4545
5420	3960	4034	4040	4151	4157	4231	4237	4348	4354	4428	4434	5112
5762	4625	4631	4742	4748	4822	4828	4939	4945	5019	5025	5109	5377
6060	5127	5156	5179	5182	5191	5197	5226	5232	5280	5286	5354	5749
6412	5420	5433	5499	5505	5555	5561	5584	5589	5640	5646	5706	6054
6784	5762	5768	5791	5834	5847	5853	5913	5956	5969	5975	5998	6389
7089	6060	6120	6163	6176	6182	6205	6248	6261	6267	6327	6370	6741
7815	6412	6455	6468	6474	6534	6577	6590	6596	6619	6625	6675	7076
8324	6784	6797	6803	6826	6869	6882	6888	6948	6991	7004	7010	7714
9107	7089	7095	7155	7198	7211	7217	7240	7283	7296	7302	7471	8317
9495	7815	7842	8042	8054	8060	8072	8157	8160	8167	8185	8247	9098
10058	8324	8342	8407	8477	8489	8567	8579	8727	8739	8859	8878	9489
10440	9107	9113	9160	9172	9178	9184	9415	9418	9424	9430	9477	10052
	9495	9501	9732	9735	9741	9747	9794	9803	9806	9812	9818	10437
	10058	10064	10111	10120	10123	10129	10135	10366	10369	10375	10381	10428
	10440	10446	10452	10683	10686	10692	10698	10745	10754	10757	10763	11000

.SINO = 020000

.SINI = 120000

.SMEH = 040000

CROSS REFERENCE TABLE -- USER SYMBOLS

.START 002402
 .SUBMC# 000040
 .SUBIC# 000340
 .SUBEC# 000360

.SO = 020000
 .TIMER = 007464
 .XOR = 000320
 .SASTA = ##### U
 .SX = 002034
 .S. = 104510

11009	11015	11063	11074	11080	11086	11317	11320	11326	11332	11379	11388	11391
11397	11405	11639	11637	11643	11649	11696	11705	11708	11714	11720	11951	11954
11960	11966	12013	12022	12025	12031	12037	12268	12277	12283	12330	12342	12348
12354	12358	12379	12500	12547	12559	12555	12571	12902	12911	12917	12964	12976
12982	12988	12979	13222	13228	13234	13281	13290	13293	13299	13305	13536	13545
13551	13558	13610	13616	13622	13738	13770	13808	14355	14367	14526		
233	732	748	1828									
	11951	12022	8702	8792	11320	11388						
	23016	23026	2503	2583	2700	2780	2897	2977	3104	3193	3330	3428
354	363	374	382	393	4019	4136	4216	4333	4413	4530	4610	4727
4807	4824	5004	5121	5191	5301	5386	5508	5593	5715	5800	5922	6007
6129	6214	6336	6421	6543	6628	6750	6835	6957	7042	7164	7249	7419
7456	7556	7636	8054	8176	8247	8333	8407	8483	8573	8733	9107	9178
9424	9495	9741	9812	10058	10129	10375	10446	10692	10763	11009	11080	11326
11397	11643	11714	11960	12031	12277	12348	12594	12665	12911	12982	13222	13228
13290	13299	13545	13616	13980	14361	14426	14613	14632				
1565	1752											
1178	1181	10437										
486	491											
2283	2294	2286	2287	2290	2291	2294	2295	2297	2298	2300	2301	2306
2307	2309	2310	2312	2313	2315	2316	2318	2319	2321	2322	2324	2325
2327	2328	2331	2332	2333	2335	2337	2338	2340	2341	2344	2345	2347
2348	2350	2351	2352	2354	2355	2357	2358	2360	2361	2366	2370	2371
2374	2375	2377	2378	2380	2381	2382	2384	2385	2386	2390	2393	2395
2396	2399	2400	2401	2402	2403	2405	2406	2407	2408	2412	2414	2415
2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2431	2434	2438
2437	2440	2441	2442	2443	2444	2445	2446	2447	2448	2451	2453	2454
2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2480	2483	2484
2487	2488	2489	2490	2491	2492	2493	2494	2495	2496	2500	2503	2504
2510	2511	2512	2513	2514	2515	2516	2517	2518	2519	2523	2526	2527
2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2544	2547	2548
2551	2552	2553	2554	2555	2556	2557	2558	2559	2560	2564	2567	2568
2571	2572	2573	2574	2575	2576	2577	2578	2579	2580	2584	2587	2588
2591	2592	2593	2594	2595	2596	2597	2598	2599	2600	2604	2607	2608
2611	2612	2613	2614	2615	2616	2617	2618	2619	2620	2624	2627	2628
2631	2632	2633	2634	2635	2636	2637	2638	2639	2640	2644	2647	2648
2651	2652	2653	2654	2655	2656	2657	2658	2659	2660	2664	2667	2668
2671	2672	2673	2674	2675	2676	2677	2678	2679	2680	2684	2687	2688
2691	2692	2693	2694	2695	2696	2697	2698	2699	2700	2704	2707	2708
2713	2715	2716	2718	2719	2721	2722	2723	2725	2726	2729	2731	2732
2734	2735	2738	2739	2741	2742	2744	2745	2746	2747	2750	2751	2756
2757	2759	2760	2761	2765	2768	2769	2771	2772	2773	2774	2775	2781
2783	2784	2786	2787	2789	2790	2792	2793	2795	2796	2798	2799	2801
2802	2805	2806	2808	2809	2811	2812	2814	2815	2816	2819	2821	2822
2824	2825	2827	2828	2829	2831	2832	2837	2838	2841	2844	2845	2847
2848	2849	2851	2852	2853	2854	2855	2856	2857	2858	2861	2862	2863
2864	2865	2867	2868	2869	2871	2872	2873	2874	2875	2879	2881	2882
2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2895	2896	2897
2898	2899	2900	2901	2902	2903	2904	2905	2906	2907	2911	2912	2913
2916	2918	2919	2920	2921	2922	2923	2924	2925	2926	2929	2931	2932
2938	2939	2941	2942	2943	2944	2945	2946	2947	2948	2951	2952	2953
2962	2965	2966	2968	2969	2971	2972	2973	2974	2975	2978	2979	2980
2986	2987	2989	2990	2992	2993	2995	2996	2997	2998	3001	3003	3004
3006	3008	3009	3011	3012	3015	3016	3018	3019	3021	3022	3023	3024
3027	3028	3033	3034	3038	3039	3041	3042	3043	3045	3046	3047	3048
3076	3079	3080	3083	3084	3086	3087	3089	3090	3091	3092	3093	3094
3098	3099	3104	3105	3107	3108	3110	3111	3113	3114	3115	3117	3119
3120	3122	3123	3125	3126	3129	3130	3132	3133	3135	3136	3138	3139

CROSS REFERENCE TABLE -- USER SYMBOLS

4544	4555	4566	4577	4588	4599	4600	4611	4622	4633	4644	4655	4666	4677	4688	4699	4700	4711	4722	4733	4744	4755	4766	4777	4788	4799	4800	4811	4822	4833	4844	4855	4866	4877	4888	4899	4900	4911	4922	4933	4944	4955	4966	4977	4988	4999	5000	5011	5022	5033	5044	5055	5066	5077	5088	5099	5100	5111	5122	5133	5144	5155	5166	5177	5188	5199	5200	5211	5222	5233	5244	5255	5266	5277	5288	5299	5300	5311	5322	5333	5344	5355	5366	5377	5388	5399	5400	5411	5422	5433	5444	5455	5466	5477	5488	5499	5500	5511	5522	5533	5544	5555	5566	5577	5588	5599	5600	5611	5622	5633	5644	5655	5666	5677	5688	5699	5700	5711	5722	5733	5744	5755	5766	5777	5788	5799	5800	5811	5822	5833	5844	5855	5866	5877	5888	5899	5900	5911	5922	5933	5944	5955	5966	5977	5988	5999	6000
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

CROSS REFERENCE TABLE -- USER SYMBOLS

5972	5973	5975	5976	5981	5982	5984	5985	5989	5990	5992	5993	5995
5996	5998	5999	6001	6002	6007	6008	6010	6011	6013	6014	6016	6017
6019	6020	6022	6023	6025	6026	6028	6029	6031	6032	6035	6036	6038
6039	6041	6042	6044	6045	6048	6049	6051	6053	6054	6055	6057	6058
6060	6061	6066	6067	6071	6072	6074	6075	6077	6078	6104	6105	6107
6108	6111	6112	6114	6115	6117	6118	6120	6121	6123	6124	6129	6130
6132	6133	6135	6136	6138	6139	6141	6142	6144	6145	6147	6148	6150
6151	6153	6154	6157	6158	6160	6161	6163	6164	6166	6167	6170	6171
6173	6174	6175	6177	6179	6180	6182	6183	6188	6189	6191	6192	6196
6197	6199	6200	6202	6203	6205	6206	6208	6209	6214	6215	6217	6218
6220	6221	6223	6224	6225	6227	6229	6230	6232	6233	6235	6236	6238
6239	6242	6243	6245	6246	6246	6249	6251	6252	6253	6255	6258	6259
6261	6262	6264	6265	6267	6268	6273	6274	6278	6279	6281	6282	6284
6285	6311	6312	6314	6315	6318	6319	6321	6322	6324	6325	6327	6328
6330	6331	6336	6337	6339	6340	6342	6343	6345	6346	6348	6349	6351
6352	6354	6355	6357	6358	6360	6361	6364	6365	6367	6368	6370	6371
6373	6374	6377	6379	6388	6381	6383	6384	6386	6387	6389	6390	6395
6396	6398	6399	6403	6404	6405	6407	6409	6410	6412	6413	6415	6416
6421	6423	6424	6425	6427	6428	6430	6431	6433	6434	6436	6437	6439
6440	6443	6443	6445	6446	6449	6450	6451	6453	6455	6456	6458	6459
6462	6463	6465	6466	6468	6469	6471	6472	6474	6475	6480	6481	6485
6486	6488	6489	6491	6492	6493	6495	6497	6498	6502	6503	6505	6509
6531	6532	6534	6535	6537	6538	6543	6544	6547	6547	6549	6550	6552
6553	6555	6556	6558	6559	6561	6562	6564	6565	6567	6568	6571	6572
6574	6575	6577	6578	6580	6581	6584	6585	6587	6588	6590	6591	6593
6594	6595	6597	6602	6603	6605	6606	6608	6610	6611	6613	6616	6617
6619	6620	6623	6623	6628	6629	6631	6632	6634	6635	6637	6638	6640
6641	6643	6644	6645	6647	6648	6650	6651	6653	6655	6657	6659	6660
6662	6663	6665	6666	6669	6670	6672	6673	6675	6676	6678	6679	6681
6682	6687	6688	6690	6693	6694	6696	6698	6699	6705	6706	6708	6709
6732	6733	6735	6736	6738	6739	6741	6742	6744	6745	6750	6751	6753
6754	6756	6757	6759	6760	6762	6763	6765	6766	6768	6769	6771	6772
6774	6775	6778	6779	6781	6782	6784	6785	6787	6788	6791	6792	6794
6795	6797	6798	6800	6801	6803	6804	6809	6810	6812	6813	6817	6818
6820	6821	6823	6824	6826	6827	6829	6830	6835	6836	6838	6839	6841
6842	6844	6845	6847	6848	6850	6851	6853	6854	6855	6857	6859	6860
6863	6864	6866	6867	6869	6870	6872	6873	6876	6877	6879	6880	6882
6883	6885	6886	6888	6889	6891	6892	6893	6899	6900	6902	6905	6906
6932	6933	6935	6936	6939	6940	6942	6943	6945	6946	6948	6949	6951
6952	6957	6958	6960	6961	6963	6964	6966	6967	6969	6970	6972	6973
6975	6976	6978	6979	6981	6982	6985	6986	6988	6989	6991	6992	6994
6995	6998	6999	7001	7002	7004	7005	7007	7008	7010	7011	7016	7017
7019	7020	7024	7025	7027	7028	7030	7031	7033	7034	7036	7037	7042
7043	7045	7046	7048	7049	7051	7052	7054	7055	7057	7058	7060	7061
7063	7064	7066	7067	7070	7071	7073	7074	7076	7077	7079	7080	7083
7084	7086	7087	7089	7090	7092	7093	7095	7096	7101	7102	7106	7107
7109	7110	7112	7113	7139	7140	7142	7143	7146	7147	7149	7150	7152
7153	7155	7156	7158	7159	7164	7165	7167	7168	7170	7171	7173	7174
7176	7177	7179	7180	7182	7183	7185	7186	7188	7189	7192	7193	7195
7196	7198	7199	7201	7202	7205	7206	7208	7209	7211	7212	7214	7215
7217	7218	7223	7224	7226	7227	7231	7232	7234	7235	7237	7238	7240
7241	7243	7244	7249	7250	7252	7253	7255	7256	7258	7259	7261	7262
7264	7265	7267	7268	7270	7271	7273	7274	7277	7278	7280	7281	7283
7284	7286	7287	7290	7291	7293	7294	7296	7297	7299	7300	7302	7303
7308	7309	7313	7314	7316	7317	7319	7320	7344	7345	7347	7348	7350
7351	7353	7354	7356	7357	7359	7360	7362	7363	7365	7366	7368	7369

CROSS REFERENCE TABLE -- USER SYMBOLS

7374	7375	7377	7378	7379	7380	7381	7382	7383	7384	7385	7386	7387	7388	7389	7390	7392
7393	7394	7395	7396	7397	7398	7399	7400	7401	7402	7403	7404	7405	7406	7407	7410	7411
7413	7414	7415	7416	7417	7418	7419	7420	7421	7422	7423	7424	7425	7426	7427	7432	7434
7436	7437	7438	7439	7440	7441	7442	7443	7444	7445	7446	7447	7448	7449	7450	7455	7457
7453	7454	7455	7456	7457	7458	7459	7460	7461	7462	7463	7464	7465	7466	7467	7476	7478
7469	7470	7471	7472	7473	7474	7475	7476	7477	7478	7479	7480	7481	7482	7483	7521	7522
7484	7485	7486	7487	7488	7489	7490	7491	7492	7493	7494	7495	7496	7497	7498	7540	7542
7499	7500	7501	7502	7503	7504	7505	7506	7507	7508	7509	7510	7511	7512	7513	7560	7561
7514	7515	7516	7517	7518	7519	7520	7521	7522	7523	7524	7525	7526	7527	7528	7582	7584
7529	7530	7531	7532	7533	7534	7535	7536	7537	7538	7539	7540	7541	7542	7543	7602	7603
7544	7545	7546	7547	7548	7549	7550	7551	7552	7553	7554	7555	7556	7557	7558	7624	7626
7559	7560	7561	7562	7563	7564	7565	7566	7567	7568	7569	7570	7571	7572	7573	7644	7645
7574	7575	7576	7577	7578	7579	7580	7581	7582	7583	7584	7585	7586	7587	7588	7666	7668
7589	7590	7591	7592	7593	7594	7595	7596	7597	7598	7599	7600	7601	7602	7603	7687	7688
7604	7605	7606	7607	7608	7609	7610	7611	7612	7613	7614	7615	7616	7617	7618	7709	7711
7619	7620	7621	7622	7623	7624	7625	7626	7627	7628	7629	7630	7631	7632	7633	7731	7732
7634	7635	7636	7637	7638	7639	7640	7641	7642	7643	7644	7645	7646	7647	7648	7774	7776
7649	7650	7651	7652	7653	7654	7655	7656	7657	7658	7659	7660	7661	7662	7663	7794	7795
7664	7665	7666	7667	7668	7669	7670	7671	7672	7673	7674	7675	7676	7677	7678	7813	7815
7679	7680	7681	7682	7683	7684	7685	7686	7687	7688	7689	7690	7691	7692	7693	7833	7834
7694	7695	7696	7697	7698	7699	7700	7701	7702	7703	7704	7705	7706	7707	7708	7853	7855
7709	7710	7711	7712	7713	7714	7715	7716	7717	7718	7719	7720	7721	7722	7723	7898	7899
7724	7725	7726	7727	7728	7729	7730	7731	7732	7733	7734	7735	7736	7737	7738	7917	7919
7739	7740	7741	7742	7743	7744	7745	7746	7747	7748	7749	7750	7751	7752	7753	7937	7938
7754	7755	7756	7757	7758	7759	7760	7761	7762	7763	7764	7765	7766	7767	7768	7959	7961
7769	7770	7771	7772	7773	7774	7775	7776	7777	7778	7779	7780	7781	7782	7783	7979	7980
7784	7785	7786	7787	7788	7789	7790	7791	7792	7793	7794	7795	7796	7797	7798	8000	8000
7799	7800	7801	7802	7803	7804	7805	7806	7807	7808	7809	7810	7811	7812	7813	8018	8019
7814	7815	7816	7817	7818	7819	7820	7821	7822	7823	7824	7825	7826	7827	7828	8037	8039
7829	7830	7831	7832	7833	7834	7835	7836	7837	7838	7839	7840	7841	7842	7843	8060	8061
7844	7845	7846	7847	7848	7849	7850	7851	7852	7853	7854	7855	7856	7857	7858	8080	8082
7859	7860	7861	7862	7863	7864	7865	7866	7867	7868	7869	7870	7871	7872	7873	8100	8101
7874	7875	7876	7877	7878	7879	7880	7881	7882	7883	7884	7885	7886	7887	7888	8146	8148
7889	7890	7891	7892	7893	7894	7895	7896	7897	7898	7899	7900	7901	7902	7903	8167	8168
7904	7905	7906	7907	7908	7909	7910	7911	7912	7913	7914	7915	7916	7917	7918	8189	8191
7919	7920	7921	7922	7923	7924	7925	7926	7927	7928	7929	7930	7931	7932	7933	8210	8211
7934	7935	7936	7937	7938	7939	7940	7941	7942	7943	7944	7945	7946	7947	7948	8233	8235
7949	7950	7951	7952	7953	7954	7955	7956	7957	7958	7959	7960	7961	7962	7963	8257	8258
7964	7965	7966	7967	7968	7969	7970	7971	7972	7973	7974	7975	7976	7977	7978	8300	8302
7979	7980	7981	7982	7983	7984	7985	7986	7987	7988	7989	7990	7991	7992	7993	8321	8322
7994	7995	7996	7997	7998	7999	8000	8001	8002	8003	8004	8005	8006	8007	8008	8343	8345
8009	8010	8011	8012	8013	8014	8015	8016	8017	8018	8019	8020	8021	8022	8023	8364	8365
8024	8025	8026	8027	8028	8029	8030	8031	8032	8033	8034	8035	8036	8037	8038	8410	8411
8039	8040	8041	8042	8043	8044	8045	8046	8047	8048	8049	8050	8051	8052	8053	8431	8435
8054	8055	8056	8057	8058	8059	8060	8061	8062	8063	8064	8065	8066	8067	8068	8474	8475
8069	8070	8071	8072	8073	8074	8075	8076	8077	8078	8079	8080	8081	8082	8083	8496	8498
8084	8085	8086	8087	8088	8089	8090	8091	8092	8093	8094	8095	8096	8097	8098	8517	8518
8099	8100	8101	8102	8103	8104	8105	8106	8107	8108	8109	8110	8111	8112	8113	8537	8538
8114	8115	8116	8117	8118	8119	8120	8121	8122	8123	8124	8125	8126	8127	8128	8561	8562
8129	8130	8131	8132	8133	8134	8135	8136	8137	8138	8139	8140	8141	8142	8143	8577	8578
8144	8145	8146	8147	8148	8149	8150	8151	8152	8153	8154	8155	8156	8157	8158	8606	8606
8159	8160	8161	8162	8163	8164	8165	8166	8167	8168	8169	8170	8171	8172	8173	8629	8629
8174	8175	8176	8177	8178	8179	8180	8181	8182	8183	8184	8185	8186	8187	8188	8673	8673
8189	8190	8191	8192	8193	8194	8195	8196	8197	8198	8199	8200	8201	8202	8203	8693	8693
8204	8205	8206	8207	8208	8209	8210	8211	8212	8213	8214	8215	8216	8217	8218	8716	8716
8219	8220	8221	8222	8223	8224	8225	8226	8227	8228	8229	8230	8231	8232	8233		
8234	8235	8236	8237	8238	8239	8240	8241	8242	8243	8244	8245	8246	8247	8248		
8249	8250	8251	8252	8253	8254	8255	8256	8257	8258	8259	8260	8261	8262	8263		
8264	8265	8266	8267	8268	8269	8270	8271	8272	8273	8274	8275	8276	8277	8278		
8279	8280	8281	8282	8283	8284	8285	8286	8287	8288	8289	8290	8291	8292	8293		
8294	8295	8296	8297	8298	8299	8300	8301	8302	8303	8304	8305	8306	8307	8308		
8309	8310	8311	8312	8313	8314	8315	8316	8317	8318	8319	8320	8321	8322	8323		
8324	8325	8326	8327	8328	8329	8330	8331	8332	8333	8334	8335	8336	8337	8338		
8339	8340	8341	8342	8343	8344	8345	8346	8347	8348	8349	8350	8351	8352	8353		
8354	8355	8356	8357	8358	8359	8360	8361	8362	8363	8364	8365	8366	8367	8368		
8369	8370	8371	8372	8373	8374	8375	8376	8377	8378	8379	8380	8381	8382	8383		
8384	8385	8386	8387	8388	8389	8390	8391	8392	8393	8394	8395	8396	8397	8398		
8399	8400	8401	8402	8403	8404	8405	8406	8407	8408	8409	8410	8411	8412	8413		
8414	8415	8416	8417	8418	8419	8420	8421	8422	8423	8424	8425	8426	8427	8428		
8429	8430	8431	8432	8433	8434	8435	8436	8437	8438	8439	8440	8441	8442	8443		
8444	8445	8446	8447	8448	8449	8450	8451	8452	8453	8454	8455	8456	8457	8458		
8459	8460	8461	8462	8463	8464	8465	8466	8467	8468	8469	8470	8471	8472	8473		
8474	8475	8476	8477	8478	8479	8480	8481	8482	8483	8484	8485	8486	8487	8488		
8489	8490	8491	8492	8493	8494	8495	8496	8497	8498	8499	8500	8501	8502	8503		
8504	8505	8506	8507	8508	8509	8510	8511	8512	8513	8514	8515	8516	8517	8518		
8519	8520	8521	8522	8523	8524	8525	8526	8527	8528	8529	8530	8531	8532	8533		
8534	8535	8536	8537	8538	8539	8540	8541	8542	8543	8544	8545	8546	8547	8548		
8549	8550	8551	8552	8553	8554	8555	8556	8557	8558	8559	8560	8561	8562	8563		
8564	8565	8566	8567	8												

CROSS REFERENCE TABLE -- USER SYMBOLS

8718	8719	8721	8722	8724	8725	8727	8728	8733	8734	8736	8737	8739
8740	8742	8743	8745	8746	8748	8749	8751	8752	8755	8756	8758	8759
8761	8762	8764	8765	8768	8769	8771	8772	8774	8775	8777	8778	8780
8781	8783	8784	8785	8787	8789	8793	8795	8796	8798	8799	8803	8804
8805	8807	8809	8810	8815	8816	8817	8819	8820	8821	8822	8823	8824
8848	8850	8851	8853	8854	8856	8857	8859	8860	8861	8862	8863	8864
8865	8867	8868	8869	8870	8871	8872	8873	8874	8875	8876	8877	8878
8899	8900	8901	8902	8903	8904	8905	8906	8907	8908	8909	8910	8911
8912	8913	8914	8915	8916	8917	8918	8919	8920	8921	8922	8923	8924
8925	8926	8927	8928	8929	8930	8931	8932	8933	8934	8935	8936	8937
8938	8939	8940	8941	8942	8943	8944	8945	8946	8947	8948	8949	8950
8951	8952	8953	8954	8955	8956	8957	8958	8959	8960	8961	8962	8963
8964	8965	8966	8967	8968	8969	8970	8971	8972	8973	8974	8975	8976
8977	8978	8979	8980	8981	8982	8983	8984	8985	8986	8987	8988	8989
8990	8991	8992	8993	8994	8995	8996	8997	8998	8999	9000	9001	9002
9003	9004	9005	9006	9007	9008	9009	9010	9011	9012	9013	9014	9015
9016	9017	9018	9019	9020	9021	9022	9023	9024	9025	9026	9027	9028
9029	9030	9031	9032	9033	9034	9035	9036	9037	9038	9039	9040	9041
9042	9043	9044	9045	9046	9047	9048	9049	9050	9051	9052	9053	9054
9055	9056	9057	9058	9059	9060	9061	9062	9063	9064	9065	9066	9067
9068	9069	9070	9071	9072	9073	9074	9075	9076	9077	9078	9079	9080
9081	9082	9083	9084	9085	9086	9087	9088	9089	9090	9091	9092	9093
9094	9095	9096	9097	9098	9099	9100	9101	9102	9103	9104	9105	9106
9107	9108	9109	9110	9111	9112	9113	9114	9115	9116	9117	9118	9119
9120	9121	9122	9123	9124	9125	9126	9127	9128	9129	9130	9131	9132
9133	9134	9135	9136	9137	9138	9139	9140	9141	9142	9143	9144	9145
9146	9147	9148	9149	9150	9151	9152	9153	9154	9155	9156	9157	9158
9159	9160	9161	9162	9163	9164	9165	9166	9167	9168	9169	9170	9171
9172	9173	9174	9175	9176	9177	9178	9179	9180	9181	9182	9183	9184
9185	9186	9187	9188	9189	9190	9191	9192	9193	9194	9195	9196	9197
9198	9199	9200	9201	9202	9203	9204	9205	9206	9207	9208	9209	9210
9211	9212	9213	9214	9215	9216	9217	9218	9219	9220	9221	9222	9223
9224	9225	9226	9227	9228	9229	9230	9231	9232	9233	9234	9235	9236
9237	9238	9239	9240	9241	9242	9243	9244	9245	9246	9247	9248	9249
9250	9251	9252	9253	9254	9255	9256	9257	9258	9259	9260	9261	9262
9263	9264	9265	9266	9267	9268	9269	9270	9271	9272	9273	9274	9275
9276	9277	9278	9279	9280	9281	9282	9283	9284	9285	9286	9287	9288
9289	9290	9291	9292	9293	9294	9295	9296	9297	9298	9299	9300	9301
9302	9303	9304	9305	9306	9307	9308	9309	9310	9311	9312	9313	9314
9315	9316	9317	9318	9319	9320	9321	9322	9323	9324	9325	9326	9327
9328	9329	9330	9331	9332	9333	9334	9335	9336	9337	9338	9339	9340
9341	9342	9343	9344	9345	9346	9347	9348	9349	9350	9351	9352	9353
9354	9355	9356	9357	9358	9359	9360	9361	9362	9363	9364	9365	9366
9367	9368	9369	9370	9371	9372	9373	9374	9375	9376	9377	9378	9379
9380	9381	9382	9383	9384	9385	9386	9387	9388	9389	9390	9391	9392
9393	9394	9395	9396	9397	9398	9399	9400	9401	9402	9403	9404	9405
9406	9407	9408	9409	9410	9411	9412	9413	9414	9415	9416	9417	9418
9419	9420	9421	9422	9423	9424	9425	9426	9427	9428	9429	9430	9431
9432	9433	9434	9435	9436	9437	9438	9439	9440	9441	9442	9443	9444
9445	9446	9447	9448	9449	9450	9451	9452	9453	9454	9455	9456	9457
9458	9459	9460	9461	9462	9463	9464	9465	9466	9467	9468	9469	9470
9471	9472	9473	9474	9475	9476	9477	9478	9479	9480	9481	9482	9483
9484	9485	9486	9487	9488	9489	9490	9491	9492	9493	9494	9495	9496
9497	9498	9499	9500	9501	9502	9503	9504	9505	9506	9507	9508	9509
9510	9511	9512	9513	9514	9515	9516	9517	9518	9519	9520	9521	9522
9523	9524	9525	9526	9527	9528	9529	9530	9531	9532	9533	9534	9535
9536	9537	9538	9539	9540	9541	9542	9543	9544	9545	9546	9547	9548
9549	9550	9551	9552	9553	9554	9555	9556	9557	9558	9559	9560	9561
9562	9563	9564	9565	9566	9567	9568	9569	9570	9571	9572	9573	9574
9575	9576	9577	9578	9579	9580	9581	9582	9583	9584	9585	9586	9587
9588	9589	9590	9591	9592	9593	9594	9595	9596	9597	9598	9599	9600
9601	9602	9603	9604	9605	9606	9607	9608	9609	9610	9611	9612	9613
9614	9615	9616	9617	9618	9619	9620	9621	9622	9623	9624	9625	9626
9627	9628	9629	9630	9631	9632	9633	9634	9635	9636	9637	9638	9639
9640	9641	9642	9643	9644	9645	9646	9647	9648	9649	9650	9651	9652
9653	9654	9655	9656	9657	9658	9659	9660	9661	9662	9663	9664	9665
9666	9667	9668	9669	9670	9671	9672	9673	9674	9675	9676	9677	9678
9679	9680	9681	9682	9683	9684	9685	9686	9687	9688	9689	9690	9691
9692	9693	9694	9695	9696	9697	9698	9699	9700	9701	9702	9703	9704
9705	9706	9707	9708	9709	9710	9711	9712	9713	9714	9715	9716	9717
9718	9719	9720	9721	9722	9723	9724	9725	9726	9727	9728	9729	9730
9731	9732	9733	9734	9735	9736	9737	9738	9739	9740	9741	9742	9743
9744	9745	9746	9747	9748	9749	9750	9751	9752	9753	9754	9755	9756
9757	9758	9759	9760	9761	9762	9763	9764	9765	9766	9767	9768	9769
9770	9771	9772	9773	9774	9775	9776	9777	9778	9779	9780	9781	9782
9783	9784	9785	9786	9787	9788	9789	9790	9791	9792	9793	9794	9795
9796	9797	9798	9799	9800	9801	9802	9803	9804	9805	9806	9807	9808
9809	9810	9811	9812	9813	9814	9815	9816	9817	9818	9819	9820	9821
9822	9823	9824	9825	9826	9827	9828	9829	9830	9831	9832	9833	9834
9835	9836	9837	9838	9839	9840	9841	9842	9843	9844	9845	9846	9847
9848	9849	9850	9851	9852	9853	9854	9855	9856	9857	9858	9859	9860
9861	9862	9863	9864	9865	9866	9867	9868	9869	9870	9871	9872	9873
9874	9875	9876	9877	9878	9879	9880	9881	9882	9883	9884	9885	9886
9887	9888	9889	9890	9891	9892	9893	9894	9895	9896	9897	9898	9899
9900	9901	9902	9903	9904	9905	9906	9907	9908	9909	9910	9911	9912
9913	9914	9915	9916	9917	9918	9919	9920	9921	9922	9923	9924	9925
9926	9927	9928	9929	9930	9931	9932	9933	9934	9935	9936	9937	9938
9939	9940	9941	9942	9943	9944	9945	9946	9947	9948	9949	9950	9951
9952	9953	9954	9955	9956	9957	9958	9959	9960	9961	9962	9963	9964
9965	9966	9967	9968	9969	9970	9971	9972	9973	9974	9975	9976	9977
9978	9979	9980	9981	9982	9983	9984	9985	9986	9987	9988	9989	9990

CROSS REFERENCE TABLE -- USER SYMBOLS

9991	9993	9994	9996	9997	9999	10000	10002	10003	10005	10006	10008	10009
10011	10012	10014	10015	10017	10018	10019	10021	10023	10024	10025	10027	10029
10030	10032	10033	10035	10036	10037	10038	10041	10043	10044	10047	10049	10050
10052	10053	10058	10059	10061	10062	10063	10065	10067	10068	10070	10071	10073
10074	10076	10077	10079	10080	10081	10082	10084	10087	10088	10090	10092	10093
10096	10097	10099	10100	10102	10103	10104	10106	10109	10110	10111	10112	10114
10115	10117	10118	10120	10121	10122	10123	10125	10128	10129	10131	10132	10134
10138	10139	10141	10142	10144	10145	10146	10148	10151	10152	10153	10155	10157
10158	10160	10161	10163	10164	10165	10166	10168	10171	10172	10174	10175	10177
10179	10180	10182	10184	10185	10186	10187	10189	10191	10192	10194	10197	10198
10200	10201	10229	10230	10232	10233	10234	10236	10239	10241	10242	10244	10245
10245	10247	10248	10249	10251	10252	10253	10255	10257	10259	10260	10262	10263
10265	10266	10268	10269	10271	10272	10273	10275	10277	10279	10280	10281	10283
10284	10286	10287	10289	10290	10291	10292	10294	10297	10299	10301	10302	10303
10304	10305	10307	10308	10310	10311	10312	10314	10316	10317	10319	10320	10322
10323	10325	10326	10328	10329	10331	10332	10334	10336	10337	10338	10340	10341
10343	10344	10346	10347	10349	10350	10351	10353	10355	10357	10360	10361	10363
10364	10366	10367	10369	10370	10371	10372	10374	10377	10381	10382	10384	10385
10387	10388	10390	10391	10393	10394	10395	10397	10400	10401	10403	10404	10405
10407	10409	10412	10413	10414	10415	10416	10417	10420	10421	10423	10425	10426
10428	10429	10432	10433	10434	10435	10436	10437	10440	10441	10443	10447	10449
10450	10452	10455	10456	10457	10458	10459	10460	10462	10464	10465	10467	10468
10471	10473	10476	10477	10478	10479	10480	10481	10483	10485	10487	10488	10490
10491	11	11	11	10497	10499	10501	10502	10504	10505	10507	10508	10511
10512	11	11	11	10518	10519	10521	10522	10524	10525	10527	10528	10531
10558	11	11	11	10554	10555	10557	10558	10560	10561	10563	10564	10565
10577	11	11	11	10583	10584	10586	10587	10589	10591	10592	10594	10595
10597	11	11	11	10593	10594	10596	10597	10599	10600	10602	10603	10604
10616	11	11	11	10617	10618	10620	10621	10623	10624	10626	10627	10628
10636	10637	11	10640	10641	10642	10643	10644	10646	10647	10649	10651	10652
10655	10657	11	10660	10661	10662	10663	10664	10666	10667	10669	10670	10673
10677	11	11	10681	10682	10683	10684	10685	10687	10688	10693	10695	10698
10699	10701	10702	10704	10705	10706	10707	10708	10710	10711	10713	10714	10716
10720	10721	10723	10724	10725	10726	10727	10728	10730	10731	10734	10736	10739
10740	10742	10743	10745	10746	10747	10748	10749	10749	10751	10754	10755	10758
10763	10764	11	10765	10766	10767	10768	10769	10772	10773	10776	10778	10781
10782	10784	11	10785	10786	10787	10788	10789	10792	10793	10797	10799	10802
10804	10805	11	10806	10807	10808	10809	10810	10811	10814	10816	10818	10822
10824	10825	11	10826	10827	10828	10829	10831	10832	10835	10836	10837	10839
10870	10872	11	10873	10874	10875	10876	10877	10878	10881	10882	10884	10888
10890	10891	11	10892	10893	10894	10895	10896	10897	10899	10900	10901	10908
10909	10911	10912	10913	10915	10917	10918	10919	10920	10921	10923	10924	10927
10929	10930	10932	10933	10935	10936	10937	10938	10939	10941	10942	10944	10947
10948	10950	10951	10953	10954	10955	10956	10957	10959	10960	10962	10963	10965
10968	10969	10971	10972	10974	10975	10976	10977	10978	10980	10981	10983	10984
10987	10990	10991	10994	10995	10997	10998	10999	11000	11001	11003	11004	10997
11012	11013	11015	11016	11018	11019	11021	11022	11024	11025	11027	11028	11030
11031	11034	11035	11037	11038	11040	11041	11043	11044	11047	11048	11050	11051
11053	11054	11056	11057	11059	11060	11062	11063	11065	11066	11068	11069	11071
11072	11074	11075	11080	11081	11083	11084	11086	11087	11089	11090	11092	11093
11095	11096	11098	11099	11101	11102	11105	11106	11108	11109	11111	11112	11114
11115	11118	11119	11121	11122	11124	11125	11127	11128	11130	11131	11133	11135
11136	11138	11139	11141	11142	11145	11146	11148	11149	11151	11152	11160	11181
11183	11184	11186	11187	11189	11190	11192	11193	11195	11196	11198	11199	11201
11202	11204	11205	11207	11208	11210	11211	11213	11214	11216	11217	11219	11220
11222	11223	11225	11226	11228	11229	11231	11232	11234	11235	11237	11238	11240

CROSS REFERENCE TABLE -- USER SYMBOLS

11241	11243	11244	11246	11247	11249	11250	11252	11253	11255	11256	11258	11259
11281	11282	11284	11286	11287	11289	11290	11291	11293	11294	11296	11297	11299
11300	11301	11303	11305	11306	11308	11311	11312	11314	11315	11317	11318	11320
11344	11345	11347	11349	11350	11352	11353	11355	11357	11358	11360	11361	11364
11365	11367	11368	11370	11371	11372	11374	11375	11377	11379	11380	11382	11383
11407	11408	11410	11412	11413	11414	11416	11418	11419	11421	11423	11424	11426
11428	11429	11431	11433	11434	11435	11437	11438	11441	11442	11444	11445	11447
11448	11450	11451	11453	11454	11455	11457	11458	11461	11462	11465	11465	11468
11469	11471	11472	11474	11475	11476	11478	11479	11481	11482	11485	11485	11488
11515	11516	11518	11520	11521	11522	11524	11525	11527	11528	11530	11531	11533
11524	11525	11527	11529	11530	11531	11533	11534	11536	11537	11539	11541	11543
11554	11555	11557	11559	11560	11561	11563	11564	11566	11567	11569	11570	11572
11573	11574	11576	11578	11579	11580	11582	11583	11585	11586	11588	11590	11591
11593	11594	11596	11598	11599	11600	11602	11603	11605	11606	11608	11609	11611
11612	11614	11615	11617	11618	11620	11621	11623	11625	11626	11628	11631	11632
11634	11635	11637	11638	11640	11641	11643	11644	11646	11647	11649	11653	11655
11656	11658	11659	11661	11662	11664	11665	11666	11669	11671	11672	11674	11675
11677	11678	11681	11682	11684	11685	11686	11688	11690	11691	11693	11694	11696
11697	11699	11700	11702	11703	11705	11706	11708	11709	11714	11715	11717	11718
11720	11721	11723	11724	11726	11727	11729	11730	11732	11733	11735	11736	11739
11740	11742	11743	11745	11746	11748	11749	11751	11753	11755	11756	11758	11759
11761	11762	11764	11765	11767	11769	11770	11772	11773	11775	11776	11779	11780
11782	11783	11785	11786	11814	11815	11817	11818	11820	11821	11823	11824	11826
11827	11829	11830	11832	11833	11835	11836	11838	11839	11841	11842	11844	11845
11847	11848	11850	11851	11853	11854	11855	11857	11859	11860	11862	11863	11865
11856	11858	11859	11871	11872	11874	11875	11877	11878	11880	11881	11883	11884
11886	11887	11889	11890	11892	11893	11894	11896	11898	11899	11901	11902	11904
11905	11907	11908	11910	11911	11913	11914	11916	11917	11919	11920	11922	11923
11925	11926	11928	11929	11931	11932	11934	11935	11937	11939	11941	11942	11945
11946	11948	11949	11951	11952	11954	11955	11957	11958	11960	11961	11962	11965
11969	11970	11972	11973	11974	11976	11978	11979	11981	11982	11984	11986	11988
11989	11991	11992	11994	11995	11997	11998	11999	12001	12002	12004	12006	12008
12010	12011	12013	12014	12015	12017	12019	12020	12022	12023	12025	12027	12028
12032	12034	12035	12037	12038	12040	12041	12043	12044	12046	12047	12049	12050
12052	12053	12055	12057	12059	12060	12061	12063	12064	12066	12067	12069	12072
12073	12075	12076	12078	12079	12081	12082	12084	12085	12087	12089	12090	12092
12093	12096	12097	12099	12100	12102	12103	12105	12106	12108	12109	12137	12138
12140	12141	12143	12144	12146	12147	12148	12150	12153	12153	12155	12156	12158
12159	12161	12162	12164	12165	12167	12168	12170	12171	12173	12174	12176	12177
12179	12180	12182	12183	12185	12186	12188	12189	12191	12192	12194	12195	12197
12198	12200	12201	12203	12204	12206	12207	12209	12210	12212	12213	12215	12216
12218	12219	12221	12222	12224	12225	12227	12228	12230	12231	12233	12234	12236
12237	12239	12240	12242	12243	12245	12246	12248	12249	12251	12252	12254	12255
12258	12259	12262	12263	12265	12266	12268	12269	12271	12272	12274	12278	12280
12281	12283	12284	12286	12287	12289	12290	12292	12293	12295	12297	12298	12299
12302	12303	12305	12306	12308	12309	12311	12312	12314	12316	12318	12319	12321
12322	12324	12325	12327	12328	12330	12331	12333	12334	12336	12337	12339	12340
12342	12343	12348	12349	12351	12352	12354	12355	12357	12358	12360	12361	12363
12364	12366	12367	12369	12370	12373	12374	12376	12377	12379	12380	12382	12383
12386	12387	12389	12390	12392	12393	12395	12396	12398	12399	12401	12403	12404
12406	12407	12409	12410	12413	12414	12416	12417	12419	12420	12448	12449	12451
12422	12424	12425	12427	12428	12430	12431	12433	12434	12436	12438	12439	12440
12438	12440	12441	12443	12445	12446	12448	12449	12451	12452	12454	12455	12457
12458	12459	12461	12462	12464	12465	12467	12468	12469	12470	12472	12473	12474
12472	12473	12475	12476	12478	12479	12481	12482	12484	12485	12487	12488	12490

CROSS REFERENCE TABLE -- USER SYMBOLS

13739	13742	13743	13745	13746	13748	13749	13751	13752	13754	13755	13757	13758
13760	13761	13763	13764	13767	13768	13770	13771	13774	13775	13777	13778	13780
13781	13783	13784	13786	13787	13789	13790	13792	13793	13795	13796	13798	13799
13801	13802	13805	13806	13808	13809	13812	13813	13815	13816	13818	13819	13822
13823	13825	13826	13828	13829	13837	13838	13840	13841	13843	13844	13846	13847
13870	13871	13873	13874	13876	13877	13879	13880	13882	13883	13885	13886	13888
13889	13892	13893	13895	13896	13900	13901	13903	13904	13906	13907	13909	13910
13912	13913	13915	13916	13918	13919	13921	13923	13927	13928	13930	13931	13933
13934	13936	13937	13939	13940	13943	13943	13945	13946	13948	13949	13951	13952
13954	13955	13957	13958	13961	13963	13965	13966	13968	13969	13971	13972	13974
13975	13980	13981	13983	13984	13987	13989	13990	13991	13993	13994	13996	13997
13999	14000	14003	14004	14006	14007	14009	14010	14012	14013	14015	14016	14018
14019	14021	14022	14024	14025	14027	14028	14031	14032	14034	14035	14037	14038
14040	14041	14046	14047	14049	14050	14053	14054	14056	14057	14059	14060	14063
14064	14066	14067	14069	14070	14073	14074	14076	14077	14079	14080	14083	14084
14086	14087	14089	14090	14093	14094	14096	14097	14099	14100	14103	14104	14106
14107	14109	14110	14113	14114	14116	14117	14121	14122	14125	14126	14128	14129
14131	14132	14135	14136	14138	14139	14141	14142	14145	14146	14148	14149	14151
14152	14155	14156	14158	14159	14161	14162	14165	14166	14168	14169	14171	14172
14175	14176	14178	14179	14181	14182	14185	14186	14188	14189	14191	14192	14195
14196	14198	14199	14204	14205	14207	14208	14211	14212	14214	14215	14217	14218
14220	14221	14223	14224	14226	14227	14230	14231	14233	14234	14236	14237	14239
14240	14243	14244	14246	14247	14249	14250	14252	14253	14256	14257	14259	14260
14262	14263	14265	14266	14269	14270	14272	14273	14275	14276	14278	14279	14282
14283	14285	14286	14288	14289	14291	14292	14295	14296	14298	14299	14301	14302
14304	14305	14308	14309	14311	14312	14314	14315	14317	14318	14321	14322	14324
14325	14327	14328	14330	14331	14334	14336	14337	14338	14343	14344	14346	14347
14349	14350	14352	14353	14356	14355	14361	14362	14364	14365	14367	14368	14370
14371	14373	14374	14375	14377	14380	14381	14383	14384	14386	14387	14389	14390
14392	14393	14395	14396	14398	14399	14401	14402	14404	14405	14408	14409	14411
14412	14414	14415	14417	14418	14420	14421	14423	14427	14429	14430	14432	14433
14436	14436	14438	14439	14441	14442	14444	14445	14448	14449	14451	14452	14455
14456	14457	14459	14461	14462	14467	14468	14470	14471	14473	14474	14476	14477
14499	14500	14502	14503	14505	14506	14507	14509	14511	14512	14514	14515	14517
14518	14520	14521	14523	14524	14525	14527	14528	14530	14532	14533	14535	14536
14538	14539	14541	14542	14544	14545	14547	14549	14551	14552	14553	14554	14556
14557	14559	14560	14562	14563	14565	14566	14569	14571	14572	14574	14575	14575
14577	14578	14580	14581	14583	14584	14586	14587	14589	14590	14592	14593	14595
14596	14598	14599	14601	14602	14604	14605	14607	14608	14610	14613	14616	14617
14619	14620	14623	14624	14626	14627	14630	14633	14635	14636	14638	14639	14641
14642	14644	14646	14647	14648	14650	14651	14654	14655	14657	14658	14660	14661
14664	14665	14667	14668	14670	14671	14674	14675	14677	14678	14680	14681	14683
14709	14711	14712	14714	14715	14717	14718	14720	14721	14723	14724	14726	14727
14729	14730	14732	14733	14735	14736	14738	14739	14741	14742	14747	14748	14750
14751	14753	14754	14758	14759	14761	14762	14764	14765	14767	14768	14770	14771
14773	14774	14776	14777	14779	14780	14784	14785	14787	14788	14809	14810	14812
14813	14815	14816	14818	14819	14821	14822	14824	14825	14827	14828	14830	14831
14833	14834	14836	14837	14839	14840	14842	14843	14845	14846	14848	14849	14851
14852	14854	14855	14857	14858	14860	14861	14863	14864	14866	14867	14869	14870
14872	14873	14875	14876	14885	14886	14888	14889	14891	14892	14894	14895	14897
14898	14900	14901	14903	14904	14906	14907	14909	14910	14912	14913	14915	14916
14918	14919	14921	14922	14924	14925	14927	14928	14930	14931	14933	14934	14936
14937	14939	14940	14942	14943	14945	14946	14948	14950	14951	14953	14954	14956
14964	14965	14967	14968	14970	14971	14973	14974	14976	14977	14979	14980	14982
14983	14985	14986	14988	14989	14991	14992	14994	14995	14997	14998	15000	15001
15010	15011	15013	15014	15016	15017	15019	15020	15022	15023	15025	15026	15028

ADDMC	11636	11704																
ADD1																		
AND																		
BB0	3091	3180	3308	3406	7860	9734	9802	14534	14835	14905	14969	14990						
BB1	7409	7616	7643	7826	7987	8870	8921	14570	14853	15036								
BB2	14576	14591	14923															
BB3	13731	13759	13785	14558	14740													
BB4	2361	2441	2558	2638	2755	2835	2952	3032	3159	3248	3385	3483	3600	3680				
BB7	3797	3877	4074	4191	4271	4388	4468	4585	4665	4782	4862	4979	5059	5167				
	5237	5359	5444	5566	5651	5858	5980	6065	6187	6272	6394	6479	6601	6686				
	6808	6993	7015	7100	7222	7307	8231	8388	13797	14546	14734	14760	14847	14856	14917			
	14926	14981	14996	15030														
BC																		
BZ																		
	3744	3941	4021	4138	4218	4335	4415	4532	4612	4729	4809	4926	5006	5123	3627			
	5193	5388	5510	5595	5717	5802	5924	6009	6131	6216	6338	6423	6545	6630	5123			
	6253	6459	6544	6666	6751	6836	6958	7043	7165	7250	7367	7452	7569	7654	6630			
	8335	8541	8626	8748	8833	8955	9040	9162	9247	9369	9454	9576	9661	9783	8178			
	10186	10392	10477	10600	10685	10807	10892	11014	11100	11222	11307	11429	11514	11636	10131			
	11771	11977	12062	12185	12270	12392	12477	12600	12685	12807	12892	13014	13100	13222	11716			
	13356	13562	13647	13770	13855	13977	14062	14185	14270	14392	14477	14600	14685	14807	13301			
CALL																		
	2833	2918	3040	3125	3247	3332	3454	3539	3661	3746	3868	3953	4075	4160				
	4245	4367	4452	4574	4659	4781	4866	4988	5073	5195	5280	5402	5487	5609				
	5580	5702	5787	5909	5994	6116	6201	6323	6408	6530	6615	6737	6822	6944				
	6915	7037	7122	7244	7329	7451	7536	7658	7743	7865	7950	8072	8157	8279				
	8364	8479	8591	8713	8798	8920	9005	9127	9212	9334	9419	9541	9626	9748				
	9833	9948	10060	10182	10267	10389	10474	10596	10681	10803	10888	11010	11095	11217				
	11302	11387	11509	11594	11716	11801	11923	12008	12130	12215	12337	12422	12544	12629				
	12721	12806	12928	13013	13135	13220	13342	13427	13549	13634	13756	13841	13963	14048				
	14133	14218	14340	14425	14547	14632	14754	14839	14961	15046	15168	15253	15375	15460				
	15545	15667	15752	15874	15959	16081	16166	16288	16373	16495	16580	16702	16787	16909				
	17074	17159	17281	17366	17488	17573	17695	17780	17902	17987	18109	18194	18316	18401				
	18486	18608	18693	18815	18900	19022	19107	19229	19314	19436	19521	19643	19728	19850				
	19935	20057	20142	20264	20349	20471	20556	20678	20763	20885	20970	21092	21177	21300				
	21385	21507	21592	21714	21799	21921	22006	22128	22213	22335	22420	22542	22627	22749				
	22834	22919	23041	23126	23248	23333	23455	23540	23662	23747	23869	23954	24076	24161				
	24246	24368	24453	24575	24660	24782	24867	24989	25074	25196	25281	25403	25488	25610				
	25695	25780	25902	25987	26109	26194	26316	26401	26523	26608	26730	26815	26937	27022				
	27107	27229	27314	27436	27521	27643	27728	27850	27935	28057	28142	28264	28349	28471				
	28556	28641	28763	28848	28970	29055	29177	29262	29384	29469	29591	29676	29798	29883				
	29968	30083	30168	30290	30375	30497	30582	30704	30789	30911	30996	31118	31203	31325				
	31390	31475	31597	31682	31804	31889	32011	32096	32218	32303	32425	32510	32632	32717				
	32839	32924	33046	33131	33253	33338	33460	33545	33667	33752	33874	33959	34081	34166				
	34283	34368	34490	34575	34697	34782	34904	34989	35111	35196	35318	35403	35525	35610				
	35732	35817	35939	36024	36146	36231	36353	36438	36560	36645	36767	36852	36974	37059				
	37171	37256	37378	37463	37585	37670	37792	37877	38000	38085	38207	38292	38414	38500				
	38611	38726	38848	38933	39055	39140	39262	39347	39469	39554	39676	39761	39883	39968				
	40090	40175	40297	40382	40504	40589	40711	40796	40918	40993	41115	41200	41322	41407				
	41529	41614	41736	41821	41943	42028	42150	42235	42357	42442	42564	42649	42771	42856				
	42978	43063	43185	43270	43392	43477	43600	43685	43807	43892	44014	44100	44222	44307				
	44429	44514	44636	44721	44843	44928	45050	45135	45257	45342	45464	45550	45672	45757				
	45879	45964	46086	46171	46293	46378	46500	46585	46707	46792	46914	47000	47122	47207				
	47329	47414	47536	47621	47743	47828	47950	48035	48157	48242	48364	48450	48572	48657				
	48779	48864	48986	49071	49193	49278	49400	49485	49607	49692	49814	49900	50022	50107				
	50234	50319	50441	50526	50648	50733	50855	50940	51062	51147	51269	51354	51476	51561				
	51683	51768	51890	51975	52097	52182	52304	52389	52511	52596	52718	52803	52925	53010				
	53132	53217	53339	53424	53546	53631	53753	53838	53960	54045	54167	54252	54374	54459				
	54581	54666	54788	54873	54995	55080	55202	55287	55409	55494	55616	55701	55823	55908				
	56030	56115	56237	56322	56444	56529	56651	56736	56858	56943	57065	57150	57272	57357				
	57479	57564	57686	57771	57893	57978	58100	58185	58307	58392	58514	58600	58722	58807				
	58929	59014	59136	59221	59343	59428	59550	59635	59757	59842	59964	60049	60171	60256				
	60378	60463	60585	60670	60792	60877	61000	61085	61207	61292	61414	61500	61622	61707				
	61829	61914	62036	62121	62243	62328	62450	62535	62657	62742	62864	62950	63072	63157				
	63279	63364	63486	63571	63693	63778	63900	63985	64107	64192	64314	64400	64522	64607				
	64729	64814	64936	65021	65143	65228	65350	65435	65557	65642	65764	65850	65972	66057				
	66179	66264	66386	66471	66593	66678	66800	66885	67007	67092	67214	67300	67422	67507				
	67629	67714	67836	67921	68043	68128	68250	68335	68457	68542	68664	68750	68872	68957				
	69079	69164	69286	69371	69493	69578	69700	69785	69907	69992	70114	70200	70322	70407				
	70529	70614	70736	70821	70943	71028	71150	71235	71357	71442								

CROSS REFERENCE TABLE -- MACRO NAMES

2603	2607	2613	2620	2626	2632	2643	2676	2679	2683	2687	2690	2693	2705	2708
2711	2714	2717	2720	2724	2730	2737	2743	2749	2763	2767	2770	2773	2785	2788
2791	2794	2797	2800	2804	2810	2817	2823	2829	2840	2873	2876	2880	2894	2897
2898	2900	2905	2908	2911	2914	2917	2923	2927	2934	2940	2946	2950	2964	2967
2970	2973	2976	2979	2984	2991	2997	3001	3007	3014	3020	3026	3037	3071	3074
3147	3153	3167	3171	3177	3177	3183	3188	3198	3201	3204	3207	3210	3213	3217
3223	3230	3236	3242	3247	3250	3254	3259	3267	3272	3279	3285	3291	3294	3297
3306	3312	3315	3318	3323	3326	3331	3335	3345	3348	3355	3361	3365	3368	3370
3378	3385	3391	3394	3398	3400	3405	3410	3417	3421	3427	3433	3437	3440	3443
3450	3456	3462	3465	3471	3474	3478	3483	3487	3492	3498	3504	3508	3511	3514
3520	3526	3532	3535	3541	3544	3548	3553	3557	3562	3568	3574	3578	3581	3584
3590	3596	3602	3605	3611	3614	3618	3623	3627	3632	3638	3644	3648	3651	3654
3660	3666	3672	3675	3681	3684	3688	3693	3699	3705	3711	3717	3723	3726	3729
3735	3741	3747	3750	3756	3759	3763	3768	3774	3780	3786	3792	3798	3804	3807
3815	3821	3827	3830	3836	3839	3845	3850	3856	3862	3868	3874	3880	3884	3887
3895	3901	3907	3912	3918	3922	3927	3932	3938	3944	3950	3956	3962	3965	3968
3975	3981	3987	3992	3998	4004	4010	4016	4022	4028	4034	4040	4046	4049	4052
4059	4065	4071	4076	4082	4088	4094	4100	4106	4112	4118	4124	4130	4133	4136
4143	4149	4155	4160	4166	4172	4178	4184	4190	4196	4202	4208	4214	4217	4220
4229	4235	4241	4247	4253	4259	4265	4271	4277	4283	4289	4295	4301	4304	4307
4316	4322	4328	4334	4340	4346	4352	4358	4364	4370	4376	4382	4388	4391	4394
4398	4404	4410	4416	4422	4428	4434	4440	4446	4452	4458	4464	4470	4473	4476
4483	4489	4495	4501	4507	4513	4519	4525	4531	4537	4543	4549	4555	4558	4561
4569	4575	4581	4587	4593	4599	4605	4611	4617	4623	4629	4635	4641	4644	4647
4656	4662	4668	4674	4680	4686	4692	4698	4704	4710	4716	4722	4728	4731	4734
4747	4753	4759	4765	4771	4777	4783	4789	4795	4801	4807	4813	4819	4822	4825
4827	4833	4839	4845	4851	4857	4863	4869	4875	4881	4887	4893	4899	4902	4905
4909	4915	4921	4927	4933	4939	4945	4951	4957	4963	4969	4975	4981	4984	4987
4995	5001	5007	5013	5019	5025	5031	5037	5043	5049	5055	5061	5067	5070	5073
5079	5085	5091	5097	5103	5109	5115	5121	5127	5133	5139	5145	5151	5154	5157
5164	5170	5176	5182	5188	5194	5200	5206	5212	5218	5224	5230	5236	5239	5242
5248	5254	5260	5266	5272	5278	5284	5290	5296	5302	5308	5314	5320	5323	5326
5332	5338	5344	5350	5356	5362	5368	5374	5380	5386	5392	5398	5404	5407	5410
5416	5422	5428	5434	5440	5446	5452	5458	5464	5470	5476	5482	5488	5491	5494
5499	5505	5511	5517	5523	5529	5535	5541	5547	5553	5559	5565	5571	5574	5577
5584	5590	5596	5602	5608	5614	5620	5626	5632	5638	5644	5650	5656	5659	5662
5668	5674	5680	5686	5692	5698	5704	5710	5716	5722	5728	5734	5740	5743	5746
5751	5757	5763	5769	5775	5781	5787	5793	5799	5805	5811	5817	5823	5826	5829
5834	5840	5846	5852	5858	5864	5870	5876	5882	5888	5894	5900	5906	5909	5912
5918	5924	5930	5936	5942	5948	5954	5960	5966	5972	5978	5984	5990	5993	5996
5999	6005	6011	6017	6023	6029	6035	6041	6047	6053	6059	6065	6071	6074	6077
6083	6089	6095	6101	6107	6113	6119	6125	6131	6137	6143	6149	6155	6158	6161
6167	6173	6179	6185	6191	6197	6203	6209	6215	6221	6227	6233	6239	6242	6245
6251	6257	6263	6269	6275	6281	6287	6293	6299	6305	6311	6317	6323	6326	6329
6335	6341	6347	6353	6359	6365	6371	6377	6383	6389	6395	6401	6407	6410	6413
6419	6425	6431	6437	6443	6449	6455	6461	6467	6473	6479	6485	6491	6494	6497
6503	6509	6515	6521	6527	6533	6539	6545	6551	6557	6563	6569	6575	6578	6581
6587	6593	6599	6605	6611	6617	6623	6629	6635	6641	6647	6653	6659	6662	6665
6671	6677	6683	6689	6695	6701	6707	6713	6719	6725	6731	6737	6743	6746	6749
6755	6761	6767	6773	6779	6785	6791	6797	6803	6809	6815	6821	6827	6830	6833
6839	6845	6851	6857	6863	6869	6875	6881	6887	6893	6899	6905	6911	6914	6917
6923	6929	6935	6941	6947	6953	6959	6965	6971	6977	6983	6989	6995	6998	7001
7007	7013	7019	7025	7031	7037	7043	7049	7055	7061	7067	7073	7079	7082	7085
7091	7097	7103	7109	7115	7121	7127	7133	7139	7145	7151	7157	7163	7166	7169
7175	7181	7187	7193	7199	7205	7211	7217	7223	7229	7235	7241	7247	7250	7253
7259	7265	7271	7277	7283	7289	7295	7301	7307	7313	7319	7325	7331	7334	7337
7343	7349	7355	7361	7367	7373	7379	7385	7391	7397	7403	7409	7415	7418	7421
7427	7433	7439	7445	7451	7457	7463	7469	7475	7481	7487	7493	7499	7502	7505

7464	7467	7470	7474	7494	7517	7520	7523	7526	7529	7532	7535	7538	7541	7544
7547	7550	7553	7559	7565	7568	7571	7574	7577	7580	7583	7586	7589	7592	7595
7598	7601	7604	7607	7613	7616	7619	7622	7625	7628	7631	7634	7637	7640	7643
7646	7649	7652	7655	7661	7664	7667	7670	7673	7676	7679	7682	7685	7688	7691
7694	7697	7700	7703	7709	7712	7715	7718	7721	7724	7727	7730	7733	7736	7739
7742	7745	7748	7751	7757	7760	7763	7766	7769	7772	7775	7778	7781	7784	7787
7790	7793	7796	7799	7805	7808	7811	7814	7817	7820	7823	7826	7829	7832	7835
7838	7841	7844	7847	7853	7856	7859	7862	7865	7868	7871	7874	7877	7880	7883
7886	7889	7892	7895	7901	7904	7907	7910	7913	7916	7919	7922	7925	7928	7931
7934	7937	7940	7943	7949	7952	7955	7958	7961	7964	7967	7970	7973	7976	7979
7982	7985	7988	7991	7997	8000	8003	8006	8009	8012	8015	8018	8021	8024	8027
8030	8033	8036	8039	8045	8048	8051	8054	8057	8060	8063	8066	8069	8072	8075
8078	8081	8084	8087	8093	8096	8099	8102	8105	8108	8111	8114	8117	8120	8123
8126	8129	8132	8135	8141	8144	8147	8150	8153	8156	8159	8162	8165	8168	8171
8174	8177	8180	8183	8189	8192	8195	8198	8201	8204	8207	8210	8213	8216	8219
8222	8225	8228	8231	8237	8240	8243	8246	8249	8252	8255	8258	8261	8264	8267
8270	8273	8276	8279	8285	8288	8291	8294	8297	8300	8303	8306	8309	8312	8315
8318	8321	8324	8327	8333	8336	8339	8342	8345	8348	8351	8354	8357	8360	8363
8366	8369	8372	8375	8381	8384	8387	8390	8393	8396	8399	8402	8405	8408	8411
8414	8417	8420	8423	8429	8432	8435	8438	8441	8444	8447	8450	8453	8456	8459
8462	8465	8468	8471	8477	8480	8483	8486	8489	8492	8495	8498	8501	8504	8507
8510	8513	8516	8519	8525	8528	8531	8534	8537	8540	8543	8546	8549	8552	8555
8558	8561	8564	8567	8573	8576	8579	8582	8585	8588	8591	8594	8597	8600	8603
8606	8609	8612	8615	8621	8624	8627	8630	8633	8636	8639	8642	8645	8648	8651
8654	8657	8660	8663	8669	8672	8675	8678	8681	8684	8687	8690	8693	8696	8699
8702	8705	8708	8711	8717	8720	8723	8726	8729	8732	8735	8738	8741	8744	8747
8750	8753	8756	8759	8765	8768	8771	8774	8777	8780	8783	8786	8789	8792	8795
8798	8801	8804	8807	8813	8816	8819	8822	8825	8828	8831	8834	8837	8840	8843
8846	8849	8852	8855	8861	8864	8867	8870	8873	8876	8879	8882	8885	8888	8891
8894	8897	8900	8903	8909	8912	8915	8918	8921	8924	8927	8930	8933	8936	8939
8942	8945	8948	8951	8957	8960	8963	8966	8969	8972	8975	8978	8981	8984	8987
8990	8993	8996	9002	9005	9008	9011	9014	9017	9020	9023	9026	9029	9032	9035
9038	9041	9044	9047	9053	9056	9059	9062	9065	9068	9071	9074	9077	9080	9083
9086	9089	9092	9095	9101	9104	9107	9110	9113	9116	9119	9122	9125	9128	9131
9134	9137	9140	9143	9149	9152	9155	9158	9161	9164	9167	9170	9173	9176	9179
9182	9185	9188	9191	9197	9200	9203	9206	9209	9212	9215	9218	9221	9224	9227
9230	9233	9236	9239	9245	9248	9251	9254	9257	9260	9263	9266	9269	9272	9275
9278	9281	9284	9287	9293	9296	9299	9302	9305	9308	9311	9314	9317	9320	9323
9326	9329	9332	9335	9341	9344	9347	9350	9353	9356	9359	9362	9365	9368	9371
9374	9377	9380	9383	9389	9392	9395	9398	9401	9404	9407	9410	9413	9416	9419
9422	9425	9428	9431	9437	9440	9443	9446	9449	9452	9455	9458	9461	9464	9467
9470	9473	9476	9479	9485	9488	9491	9494	9497	9500	9503	9506	9509	9512	9515
9518	9521	9524	9527	9533	9536	9539	9542	9545	9548	9551	9554	9557	9560	9563
9566	9569	9572	9575	9581	9584	9587	9590	9593	9596	9599	9602	9605	9608	9611
9614	9617	9620	9623	9629	9632	9635	9638	9641	9644	9647	9650	9653	9656	9659
9662	9665	9668	9671	9677	9680	9683	9686	9689	9692	9695	9698	9701	9704	9707
9710	9713	9716	9719	9725	9728	9731	9734	9737	9740	9743	9746	9749	9752	9755
9758	9761	9764	9767	9773	9776	9779	9782	9785	9788	9791	9794	9797	9800	9803
9806	9809	9812	9815	9821	9824	9827	9830	9833	9836	9839	9842	9845	9848	9851
9854	9857	9860	9863	9869	9872	9875	9878	9881	9884	9887	9890	9893	9896	9899
9902	9905	9908	9911	9917	9920	9923	9926	9929	9932	9935	9938	9941	9944	9947
9950	9953	9956	9959	9965	9968	9971	9974	9977	9980	9983	9986	9989	9992	9995
9998	10001	10004	10007	10013	10016	10019	10022	10025	10028	10031	10034	10037	10040	10043
10046	10049	10052	10055	10061	10064	10067	10070	10073	10076	10079	10082	10085	10088	10091
10094	10097	10100	10103	10109	10112	10115	10118	10121	10124	10127	10130	10133	10136	10139
10142	10145	10148	10151	10157	10160	10163	10166	10169	10172	10175	10178	10181	10184	10187
10190	10193	10196	10199	10205	10208	10211	10214	10217	10220	10223	10226	10229	10232	10235
10238	10241	10244	10247	10253	10256	10259	10262	10265	10268	10271	10274	10277	10280	10283
10286	10289	10292	10295	10301	10304	10307	10310	10313	10316	10319	10322	10325	10328	10331
10334	10337	10340	10343	10349	10352	10355	10358	10361	10364	10367	10370	10373	10376	10379
10382	10385	10388	10391	10397	10400	10403	10406	10409	10412	10415	10418	10421	10424	10427
10430	10433	10436	10439	10445	10448	10451	10454	10457	10460	10463	10466	10469	10472	10475
10478	10481	10484	10487	10493	10496	10499	10502	10505	10508	10511	10514	10517	10520	10523
10526	10529	10532	10535	10541	10544	10547	10550	10553	10556	10559	10562	10565	10568	10571
10574	10577	10580	10583	10589	10592	10595	10598	10601	10604	10607	10610	10613	10616	10619
10622	10625	10628	10631	10637	10640	10643	10646	10649	10652	10655	10658	10661	10664	10667
10670	10673	10676	10679	10685	10688	10691	10694	10697	10700	10703	10706	10709	10712	10715
10718	10721	10724	10727	10733	10736	10739	10742	10745	10748	10751	10754	10757	10760	10763
10766	10769	10772	10775	10781	10784	10787	10790	10793	10796	10799	10802	10805	10808	10811
10814	10817	10820	10823	10829	10832	10835	10838	10841	10844	10847	10850	10853	10856	10859
10862	10865	10868	10871	10877	10880	10883	10886	10889	10892	10895	10898	10901	10904	10907
10910	10913	10916	10919	10925	10928	10931	10934	10937	10940	10943	10946	10949	10952	10955
10958	10961	10964	10967	10973	10976	10979	10982	10985	10988	10991	10994	10997	11000	11003
11006	11009	11012	11015	11021	11024	11027	11030	11033	11036	11039	11042	11045	11048	11051
11054	11057	11060	11063	11069	11072	11075	11078	11081	11084	11087	11090	11093	11096	11099
11102	11105	11108	11111	11117	11120	11123	11126	11129	11132	11135	11138	11141	11144	11147
11150	11153	11156	11159	11165	11168	11171	11174	11177	11180	11183	11186	11189	11192	11195
11198	11201	11204	11207	11213	11216	11219	11222	11225	11228	11231	11234	11237	11240	11243
11246	11249	11252	11255	11261	11264	11267	11270	11273	11276	11279	11282	11285	11288	11291
11294	11297	11300	11303	11309	11312	11315	11318	11321	11324	11327	11330	11333	11336	11339
11342	11345	11348	11351	11357	11360	11363	11366	11369	11372	11375	11378	11381	11384	11387
11390	11393	11396	11399	11405	11408	11411	11414	11417	11420	11423	11426	11429	11432	11435
11438	11441	11444	11447	11453	11456	11459	11462	11465	11468	11471	11474	11477	11480	11483
11486	11489	11492	11495	11501										

CROSS REFERENCE TABLE -- MACRO NAMES

SCOPE	71#	15166													
SETPRI	176#														
SETTRA	1544#	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567	1568
SETUP	1569														
SIFBRT	176#														
	1#	2358	2438	2555	2635	2752	2832	2949	3029	3156	3245	3382	3480	3597	3677
	3794	3874	3991	4071	4188	4268	4385	4465	4582	4662	4779	4859	4976	5056	5356
	5441	5563	5648	5770	5855	5977	6062	6184	6269	6391	6476	6598	6683	6805	6890
	7012	7097	7219	7304	7370	7562	7948	8228	8385						
SKIP	176#														
SLASH	176#														
SPACE	176#														
STARS	176#	219	242	294	297	483	485	492	961	1028	1097	1176	1235	1241	1270
	1338	1523	1664	1680											
SUBMC	1#	11953	12021												
SUB1	1#														
SUB1C	1#	8541	8625	8701	8791	11319	11387								
SUB2C	1#	2305	2385	2502	2582	2699	2779	2896	2976	3103	3192	3329	3427	3544	3624
	3741	3821	3938	4018	4135	4215	4332	4412	4529	4609	4726	4806	4923	5003	5120
	5190	5300	5385	5507	5592	5714	5799	5921	6006	6128	6213	6335	6420	6542	6627
	6749	6834	6956	7041	7163	7248	7418	7455	7655	7695	8053	8175	8246	8332	8406
	8482	8572	8732	9106	9177	9423	9494	9740	9811	10057	10128	10374	10445	10691	10762
	11008	11079	11325	11396	11642	11713	11959	12030	12276	12347	12593	12664	12910	12981	13221
	13227	13289	13298	13544	13615	13979	14360	14425	14612	14631					
SMRSU	176#														
TRMTRP	1544#														
TYPBIN	176#														
TYPOEC	176#														
TYPNAM	176#														
TY-NUM	176#														
TYPOCS	176#														
TYPOCT	176#														
TYPTXT	176#														
SADC	1#	5161	5231	8008	8096	8237	8397	8532	8616	8695	8782	8864	8915	12904	12972
	13791	13967	14769	14772											
SADD	1#	9094	9165	9230	9411	9482	9547	9728	9799	9864	10045	10116	10181	10362	10433
	10498	10679	10685	10750	10753	10815	10996	11067	11132	11313	11384	11449	11530	11701	11766
	11947	12018	12083	12264	12335	12400	12581	12652	12717	12898	12969	13034	13215	13286	13351
	13532	13603	13668	14862	14932										
SALUTO	30#	8811													
SALUTI	30#	8934	9251	9568	9885	10202	10519	10836	11153	11470	11787	12104	12421	12738	13055
	13372														
SASL	1#	12587	12655												
SAUTO	1#	817													
SBEGIN	1#	2281	2478	2675	2872	3070	3287	3520	3717	3914	4111	4308	4505	4702	4899
	5097	5274	5481	5688	5895	6102	6309	6516	6723	6930	7137	7343	7517	7760	7897
	8137	8294	8455	8668	8834	8960	9277	9594	9911	10228	10545	10862	11179	11496	11813
	12130	12447	12764	13081	13398	13713	13856	14480	14698	14806					
SBR	1#	2333	2339	2346	2352	2364	2413	2419	2426	2432	2449	2452	2530	2536	2543
	2549	2561	2610	2616	2623	2629	2646	2649	2727	2733	2740	2746	2758	2807	2813
	2820	2826	2843	2846	2824	2930	2937	2949		3004	3010	3017	3023	3040	3043
	3131	3137	3144	3150	3162	3220	3226			3256	3259	3357	3363	3370	3376
	3388	3455	3461	3468	3474	3491	3494			3585	3591	3603	3652	3658	3665
	3671	3688	3691	3769	3775	3782	3788			3855	3862	3868	3885	3888	3966
	3972	3979	3985	3997	4046	4052	4059			4085	4163	4169	4176	4182	4194
	4243	4249	4256	4262	4279	4282	4360			4379	4391	4440	4446	4453	4459

\$IFLOS	10														
\$IFNE	10	14609													
\$INC	10	7829	8002	8005	8090	8093	8234	8394	8529	8613	8692	8779	8861	8909	12270
	12338	13788	13964	14011	14388	14766	14808	14811	14884	14887					
\$LAST	10														
\$LOC	10	2281	2478	2675	2872	3070	3287	3520	3717	3914	4111	4308	4505	4702	4899
	5097	5274	5481	5688	5895	6102	6309	6516	6723	6930	7137	7343	7517	7760	7897
	8137	8294	8455	8668	8834	8960	9277	9594	9911	10228	10545	10862	11179	11496	11813
	12130	12447	12764	13081	13398	13713	13856	14480	14698	14807					
\$MCLK	29														
\$MARI	10														
\$MEMD	21	8115													
\$MEMI	19	8272													
\$MEMR	22	8432													
\$MEMO	23	8645													
\$MFLT	18	8137	8294												
\$MOCK	10														
\$MPC	10	1769													
\$MUSE	30	13830													
\$MPCG	28														
\$MPC1	25	7321													
\$MPC2	26	7494													
\$MPC3	27	7873													
\$MPC4	8	7737													
\$MPC5	10	967													
\$MPC6	10	1659													
\$MPC7	29	13689													
\$MPC8	6														
\$MPC9	9	5073													
\$MPC10	30	14673													
\$MPC11	10	1967	1979	1987	2044	2052									
\$MPC12	10	1713													
\$MPC13	10	1716	1719	1756	1761										
\$MPC14	10	5158	5228	11002	11070										
\$MPC15	10	363													
\$MPC16	10	1039													
\$MPC17	10	1078													
\$MPC18	10														
\$MPC19	10	1789													
\$MPC20	12	5281	5366	5488	5573	5695	5780	5902	5987	6109	6194	6316	6401	6523	6608
	6730	6815	6937	7022	7144	7229									
\$MPC21	13	5251	5458	5665	5872	6079	6286	6493	6700	6907	7114				
\$MPC22	14														
\$MPC23	10														
\$MPC24	10	2269	2466	2663	2860	3058	3275	3508	3705	3902	4099	4296	4493	4690	4887
	5085	5262	5469	5676	5883	6090	6297	6504	6711	6918	7125	7331	7505	7748	7885
	8125	8282	8443	8656	8822	8948	9265	9582	9899	10216	10533	10850	11167	11484	11801
	12118	12435	12752	13069	13386	13701	13844	14468	14686						
\$MPC25	10	1687													
\$MPC26	10	235													
\$MPC27	10	10368	10436												
\$MPC28	10	2259	2266	2456	2463	2653	2660	2850	2857	3047	3055	3263	3272	3498	3505
	3695	3702	3892	3899	4089	4096	4286	4293	4483	4490	4680	4687	4877	4884	5074
	5082	5252	5259	5459	5466	5666	5673	5873	5880	6080	6087	6287	6294	6494	6501
	6701	6708	6908	6915	7115	7122	7322	7328	7495	7502	7738	7745	7874	7882	8116
	8122	8273	8279	8433	8440	8646	8653	8812	8819	8935	8945	9252	9262	9569	9579

CROSS REFERENCE TABLE -- MACRO NAMES

13340	13346	13357	13360	13367	13370	13374	13377	13379	13386	13392	13619	13644	13650	13657
13663	13674	13677	13684	13687	13689	13692	13695	13697	13703	13777	13780	13786	13798	13801
13815	13818	13827	13829	13830	13831	13832	13833	13834	13835	14018	14019	14040	14086	14087
14076	14080	14087	14088	14089	14090	14091	14092	14093	14094	14178	14180	14198	14214	14233
14246	14249	14257	14258	14259	14260	14261	14262	14263	14264	14279	14280	14308	14401	14404
14411	14414	14427	14428	14429	14430	14431	14432	14433	14434	14509	14510	14516	14619	14635
14657	14660	14667	14668	14669	14670	14671	14672	14673	14674	14788	14789	14854	14857	14875
14718	14723	14737	14738	14739	14740	14741	14742	14743	14744	14805	14806	14854	14857	14875
15118	15123	15137	15138	15139	15140	15141	15142	15143	15144	15219	15220	15284	15287	15288
15237	15242	15256	15257	15258	15259	15260	15261	15262	15263	15338	15339	15402	15405	15406
15377	15382	15396	15397	15398	15399	15400	15401	15402	15403	15478	15479	15542	15545	15546
15457	15462	15476	15477	15478	15479	15480	15481	15482	15483	15558	15559	15622	15625	15626
15517	15522	15536	15537	15538	15539	15540	15541	15542	15543	15618	15619	15682	15685	15686
15577	15582	15596	15597	15598	15599	15600	15601	15602	15603	15678	15679	15742	15745	15746
15637	15642	15656	15657	15658	15659	15660	15661	15662	15663	15738	15739	15802	15805	15806
15677	15682	15696	15697	15698	15699	15700	15701	15702	15703	15778	15779	15842	15845	15846
15717	15722	15736	15737	15738	15739	15740	15741	15742	15743	15818	15819	15882	15885	15886
15757	15762	15776	15777	15778	15779	15780	15781	15782	15783	15858	15859	15922	15925	15926
15797	15802	15816	15817	15818	15819	15820	15821	15822	15823	15898	15899	15962	15965	15966
15837	15842	15856	15857	15858	15859	15860	15861	15862	15863	15938	15939	16002	16005	16006
15877	15882	15896	15897	15898	15899	15900	15901	15902	15903	16008	16009	16072	16075	16076
15917	15922	15936	15937	15938	15939	15940	15941	15942	15943	16018	16019	16082	16085	16086
15957	15962	15976	15977	15978	15979	15980	15981	15982	15983	16058	16059	16122	16125	16126
15997	16002	16016	16017	16018	16019	16020	16021	16022	16023	16098	16099	16162	16165	16166
16037	16042	16056	16057	16058	16059	16060	16061	16062	16063	16138	16139	16202	16205	16206
16077	16082	16096	16097	16098	16099	16100	16101	16102	16103	16178	16179	16242	16245	16246
16117	16122	16136	16137	16138	16139	16140	16141	16142	16143	16218	16219	16282	16285	16286
16157	16162	16176	16177	16178	16179	16180	16181	16182	16183	16258	16259	16322	16325	16326
16197	16202	16216	16217	16218	16219	16220	16221	16222	16223	16298	16299	16362	16365	16366
16237	16242	16256	16257	16258	16259	16260	16261	16262	16263	16338	16339	16402	16405	16406
16277	16282	16296	16297	16298	16299	16300	16301	16302	16303	16378	16379	16442	16445	16446
16297	16302	16316	16317	16318	16319	16320	16321	16322	16323	16398	16399	16462	16465	16466
16317	16322	16336	16337	16338	16339	16340	16341	16342	16343	16418	16419	16482	16485	16486
16337	16342	16356	16357	16358	16359	16360	16361	16362	16363	16438	16439	16502	16505	16506
16357	16362	16376	16377	16378	16379	16380	16381	16382	16383	16458	16459	16522	16525	16526
16377	16382	16396	16397	16398	16399	16400	16401	16402	16403	16478	16479	16542	16545	16546
16397	16402	16416	16417	16418	16419	16420	16421	16422	16423	16498	16499	16562	16565	16566
16417	16422	16436	16437	16438	16439	16440	16441	16442	16443	16518	16519	16582	16585	16586
16437	16442	16456	16457	16458	16459	16460	16461	16462	16463	16538	16539	16602	16605	16606
16457	16462	16476	16477	16478	16479	16480	16481	16482	16483	16558	16559	16622	16625	16626
16477	16482	16496	16497	16498	16499	16500	16501	16502	16503	16578	16579	16642	16645	16646
16497	16502	16516	16517	16518	16519	16520	16521	16522	16523	16598	16599	16662	16665	16666
16517	16522	16536	16537	16538	16539	16540	16541	16542	16543	16618	16619	16682	16685	16686
16537	16542	16556	16557	16558	16559	16560	16561	16562	16563	16638	16639	16702	16705	16706
16557	16562	16576	16577	16578	16579	16580	16581	16582	16583	16658	16659	16722	16725	16726
16577	16582	16596	16597	16598	16599	16600	16601	16602	16603	16678	16679	16742	16745	16746
16597	16602	16616	16617	16618	16619	16620	16621	16622	16623	16698	16699	16762	16765	16766
16617	16622	16636	16637	16638	16639	16640	16641	16642	16643	16718	16719	16782	16785	16786
16637	16642	16656	16657	16658	16659	16660	16661	16662	16663	16738	16739	16802	16805	16806
16657	16662	16676	16677	16678	16679	16680	16681	16682	16683	16758	16759	16822	16825	16826
16677	16682	16696	16697	16698	16699	16700	16701	16702	16703	16778	16779	16842	16845	16846
16697	16702	16716	16717	16718	16719	16720	16721	16722	16723	16798	16799	16862	16865	16866
16717	16722	16736	16737	16738	16739	16740	16741	16742	16743	16818	16819	16882	16885	16886
16737	16742	16756	16757	16758	16759	16760	16761	16762	16763	16838	16839	16902	16905	16906
16757	16762	16776	16777	16778	16779	16780	16781	16782	16783	16858	16859	16922	16925	16926
16777	16782	16796	16797	16798	16799	16800	16801	16802	16803	16878	16879	16942	16945	16946
16797	16802	16816	16817	16818	16819	16820	16821	16822	16823	16898	16899	17002	17005	17006
16817	16822	16836	16837	16838	16839	16840	16841	16842	16843	16918	16919	17022	17025	17026
16837	16842	16856	16857	16858	16859	16860	16861	16862	16863	16938	16939	17042	17045	17046
16857	16862	16876	16877	16878	16879	16880	16881	16882	16883	16958	16959	17062	17065	17066
16877	16882	16896	16897	16898	16899	16900	16901	16902	16903	16978	16979	17082	17085	17086
16897	16902	16916	16917	16918	16919	16920	16921	16922	16923	17048	17049	17112	17115	17116
16917	16922	16936	16937	16938	16939	16940	16941	16942	16943	17068	17069	17132	17135	17136
16937	16942	16956	16957	16958	16959	16960	16961	16962	16963	17088	17089	17152	17155	17156
16957	16962	16976	16977	16978	16979	16980	16981	16982	16983	17108	17109	17172	17175	17176
16977	16982	16996	16997	16998	16999	17000	17001	17002	17003	17128	17129	17192	17195	17196
16997	17002	17016	17017	17018	17019	17020	17021	17022	17023	17148	17149	17212	17215	17216
17017	17022	17036	17037	17038	17039	17040	17041	17042	17043	17168	17169	17232	17235	17236
17037	17042	17056	17057	17058	17059	17060	17061	17062	17063	17188	17189	17252	17255	17256
17057	17062	17076	17077	17078	17079	17080	17081	17082	17083	17208	17209	17272	17275	17276
17077	17082	17096	17097	17098	17099	17100	17101	17102	17103	17228	17229	17292	17295	17296
17097	17102	17116	17117	17118	17119	17120	17121	17122	17123	17248	17249	17312	17315	17316
17117	17122	17136	17137	17138	17139	17140	17141	17142	17143	17268	17269	17332	17335	17336
17137	17142	17156	17157	17158	17159	17160	17161	17162	17163	17288	17289	17352	17355	17356
17157	17162	17176	17177	17178	17179	17180	17181	17182	17183	17308	17309	17372	17375	17376
17177	17182	17196	17197	17198	17199	17200	17201	17202	17203	17328	17329	17392	17395	17396
17197	17202	17216	17217	17218	17219	17220	17221	17222	17223	17348	17349	17412	17415	17416
17217	17222	17236	17237	17238	17239	17240	17241	17242	17243	17368	17369	17432	17435	17436
17237	17242	17256	17257	1										

CROSS REFERENCE TABLE -- MACRO NAMES

10033	6228
10108	6229
10173	6230
10268	6231
10043	6232
10111	6233
10179	6234
10271	6235
10049	6236
10114	6237
10194	6238
10274	6239
10052	6240
10120	6241
10229	6242
10277	6243
10058	6244
10123	6245
10232	6246
10280	6247
10064	6248
10129	6249
10235	6250
10283	6251
10067	6252
10135	6253
10238	6254
10286	6255
10070	6256
10138	6257
10241	6258
10289	6259
10073	6260
10141	6261
10244	6262
10292	6263
10076	6264
10144	6265
10247	6266
10295	6267
10079	6268
10147	6269
10253	6270
10298	6271
10083	6272
10150	6273
10256	6274
10301	6275
10089	6276
10154	6277
10259	6278
10304	6279
10096	6280
10160	6281
10262	6282
10307	6283
10099	6284
10163	6285
10265	6286
10310	6287
10102	6288
10167	6289
10265	6290
10310	6291

CROSS REFERENCE TABLE -- MACRO NAMES

10313	10316	10319	10322	10325	10328	10331	10334	10337	10340	10343	10346	10349	10352	10356
10413	10416	10419	10422	10425	10428	10431	10434	10437	10440	10443	10446	10449	10452	10456
10513	10516	10519	10522	10525	10528	10531	10534	10537	10540	10543	10546	10549	10552	10556
10613	10616	10619	10622	10625	10628	10631	10634	10637	10640	10643	10646	10649	10652	10656
10713	10716	10719	10722	10725	10728	10731	10734	10737	10740	10743	10746	10749	10752	10756
10813	10816	10819	10822	10825	10828	10831	10834	10837	10840	10843	10846	10849	10852	10856
10913	10916	10919	10922	10925	10928	10931	10934	10937	10940	10943	10946	10949	10952	10956
11013	11016	11019	11022	11025	11028	11031	11034	11037	11040	11043	11046	11049	11052	11056
11113	11116	11119	11122	11125	11128	11131	11134	11137	11140	11143	11146	11149	11152	11156
11213	11216	11219	11222	11225	11228	11231	11234	11237	11240	11243	11246	11249	11252	11256
11313	11316	11319	11322	11325	11328	11331	11334	11337	11340	11343	11346	11349	11352	11356
11413	11416	11419	11422	11425	11428	11431	11434	11437	11440	11443	11446	11449	11452	11456
11513	11516	11519	11522	11525	11528	11531	11534	11537	11540	11543	11546	11549	11552	11556
11613	11616	11619	11622	11625	11628	11631	11634	11637	11640	11643	11646	11649	11652	11656
11713	11716	11719	11722	11725	11728	11731	11734	11737	11740	11743	11746	11749	11752	11756
11813	11816	11819	11822	11825	11828	11831	11834	11837	11840	11843	11846	11849	11852	11856
11913	11916	11919	11922	11925	11928	11931	11934	11937	11940	11943	11946	11949	11952	11956
12013	12016	12019	12022	12025	12028	12031	12034	12037	12040	12043	12046	12049	12052	12056
12113	12116	12119	12122	12125	12128	12131	12134	12137	12140	12143	12146	12149	12152	12156
12213	12216	12219	12222	12225	12228	12231	12234	12237	12240	12243	12246	12249	12252	12256
12313	12316	12319	12322	12325	12328	12331	12334	12337	12340	12343	12346	12349	12352	12356
12413	12416	12419	12422	12425	12428	12431	12434	12437	12440	12443	12446	12449	12452	12456
12513	12516	12519	12522	12525	12528	12531	12534	12537	12540	12543	12546	12549	12552	12556
12613	12616	12619	12622	12625	12628	12631	12634	12637	12640	12643	12646	12649	12652	12656
12713	12716	12719	12722	12725	12728	12731	12734	12737	12740	12743	12746	12749	12752	12756
12813	12816	12819	12822	12825	12828	12831	12834	12837	12840	12843	12846	12849	12852	12856
12913	12916	12919	12922	12925	12928	12931	12934	12937	12940	12943	12946	12949	12952	12956
13013	13016	13019	13022	13025	13028	13031	13034	13037	13040	13043	13046	13049	13052	13056
13113	13116	13119	13122	13125	13128	13131	13134	13137	13140	13143	13146	13149	13152	13156
13213	13216	13219	13222	13225	13228	13231	13234	13237	13240	13243	13246	13249	13252	13256
13313	13316	13319	13322	13325	13328	13331	13334	13337	13340	13343	13346	13349	13352	13356
13413	13416	13419	13422	13425	13428	13431	13434	13437	13440	13443	13446	13449	13452	13456
13513	13516	13519	13522	13525	13528	13531	13534	13537	13540	13543	13546	13549	13552	13556
13613	13616	13619	13622	13625	13628	13631	13634	13637	13640	13643	13646	13649	13652	13656
13713	13716	13719	13722	13725	13728	13731	13734	13737	13740	13743	13746	13749	13752	13756
13813	13816	13819	13822	13825	13828	13831	13834	13837	13840	13843	13846	13849	13852	13856
13913	13916	13919	13922	13925	13928	13931	13934	13937	13940	13943	13946	13949	13952	13956

CROSS REFERENCE TABLE -- MACRO NAMES

5910	5913	5916	5928	5931	5934	5937	5940	5943	5946	5950	5956	5963	5969	5975
5999	5999	5999	5999	6001	6013	6016	6019	6022	6025	6028	6031	6035	6041	6048
6054	6054	6071	6104	6107	6111	6114	6117	6120	6123	6135	6138	6141	6144	6147
6150	6150	6157	6163	6170	6176	6182	6186	6199	6202	6205	6208	6220	6223	6226
6239	6239	6235	6238	6243	6248	6255	6261	6267	6278	6311	6314	6318	6321	6324
6327	6330	6342	6345	6348	6351	6354	6357	6360	6364	6370	6377	6383	6389	6403
6406	6409	6412	6415	6427	6430	6433	6436	6439	6442	6445	6449	6455	6462	6468
6474	6485	6518	6521	6525	6528	6531	6534	6537	6549	6552	6555	6558	6561	6564
6567	6571	6577	6584	6590	6596	6610	6613	6616	6619	6622	6624	6637	6640	6643
6646	6649	6652	6656	6662	6669	6675	6681	6692	6725	6728	6732	6735	6738	6741
6744	6756	6759	6762	6765	6768	6771	6774	6778	6784	6791	6797	6803	6817	6820
6823	6826	6829	6841	6844	6847	6850	6853	6856	6859	6863	6869	6876	6882	6888
6899	6932	6935	6939	6942	6945	6948	6951	6953	6966	6969	6972	6975	6978	6981
6985	6991	6998	7004	7010	7024	7027	7030	7033	7036	7048	7051	7054	7057	7060
7063	7066	7070	7076	7083	7089	7095	7106	7139	7142	7146	7149	7152	7155	7158
7170	7173	7176	7179	7182	7185	7188	7192	7198	7205	7211	7217	7231	7234	7237
7240	7243	7255	7258	7261	7264	7267	7270	7273	7277	7283	7290	7296	7302	7313
7344	7347	7350	7353	7356	7359	7362	7365	7368	7374	7377	7380	7383	7386	7389
7392	7395	7398	7401	7404	7407	7413	7425	7428	7431	7434	7437	7441	7450	7462
7465	7468	7471	7475	7485	7518	7521	7524	7527	7530	7533	7536	7539	7542	7545
7548	7551	7557	7560	7566	7569	7572	7575	7578	7581	7584	7587	7590	7593	7596
7599	7602	7605	7608	7614	7617	7626	7629	7632	7635	7638	7641	7647	7650	7662
7665	7668	7671	7674	7678	7687	7690	7702	7705	7708	7711	7714	7718	7728	7761
7764	7767	7770	7773	7776	7779	7782	7785	7788	7791	7794	7797	7800	7803	7806
7809	7812	7818	7821	7824	7839	7842	7846	7855	7858	7865	7898	7901	7904	7907
7910	7913	7916	7919	7922	7925	7928	7931	7934	7937	7940	7943	7946	7961	7964
7967	7970	7973	7976	7979	7982	7985	7991	7994	8015	8018	8021	8024	8027	8030
8033	8036	8039	8042	8045	8048	8060	8063	8066	8069	8072	8076	8104	8139	8142
8145	8148	8151	8154	8157	8160	8164	8167	8170	8182	8185	8188	8191	8194	8197
8201	8207	8214	8220	8226	8241	8254	8257	8264	8296	8299	8302	8305	8308	8311
8314	8317	8321	8324	8327	8339	8342	8345	8348	8351	8354	8358	8364	8371	8377
8383	8401	8414	8417	8424	8456	8459	8462	8465	8468	8471	8474	8477	8489	8492
8495	8498	8501	8504	8507	8511	8521	8526	8536	8552	8558	8561	8564	8567	8579
8582	8585	8588	8591	8595	8605	8620	8637	8639	8669	8672	8675	8681	8684	8687
8690	8712	8715	8718	8721	8724	8727	8739	8742	8745	8748	8751	8755	8761	8768
8774	8786	8803	8835	8838	8841	8844	8847	8850	8853	8856	8859	8868	8875	8878
8882	8892	8904	8907	8919	8926	8941	8961	8964	8970	8973	8976	8979	8985	8988
8991	8994	8997	9000	9003	9006	9009	9012	9015	9018	9021	9024	9027	9030	9033
9036	9039	9042	9045	9048	9051	9054	9057	9060	9063	9066	9069	9072	9075	9078
9081	9084	9088	9092	9098	9101	9104	9107	9119	9122	9125	9128	9132	9138	9145
9151	9157	9160	9163	9169	9172	9184	9187	9190	9193	9196	9199	9203	9209	9216
9222	9228	9243	9278	9281	9284	9287	9290	9293	9296	9302	9305	9308	9311	9314
9317	9320	9323	9326	9329	9332	9335	9338	9341	9344	9347	9350	9353	9356	9359
9362	9365	9368	9371	9374	9377	9380	9383	9386	9389	9392	9395	9398	9401	9405
9409	9415	9430	9433	9436	9439	9442	9445	9449	9455	9462	9468	9474	9477	9480
9489	9501	9504	9507	9510	9513	9516	9520	9526	9533	9539	9545	9560	9585	9598
9601	9604	9607	9610	9613	9619	9622	9625	9628	9631	9634	9637	9640	9643	9646
9649	9652	9655	9658	9661	9664	9667	9670	9673	9676	9679	9682	9685	9688	9691
9694	9697	9700	9703	9706	9709	9712	9715	9718	9722	9726	9732	9747	9750	9753
9756	9759	9762	9766	9772	9779	9785	9791	9794	9797	9806	9818	9821	9824	9827
9830	9833	9837	9843	9850	9856	9862	9877	9912	9915	9918	9921	9924	9927	9930
9936	9939	9942	9945	9948	9951	9954	9957	9960	9963	9966	9969	9972	9975	9978
9981	9984	9987	9990	9993	9996	9999	10002	10005	10008	10011	10014	10017	10020	10023
10026	10029	10032	10035	10039	10043	10049	10054	10067	10070	10073	10076	10079	10083	10089
10096	10102	10108	10111	10114	10123	10135	10138	10141	10144	10147	10150	10154	10160	10167
10173	10179	10194	10229	10232	10235	10238	10241	10244	10247	10253	10256	10259	10262	10265

CROSS REFERENCE TABLE -- MACRO NAMES

10268	10271	10274	10277	10280	10283	10286	10289	10292	10295	10298	10301	10304	10307	10310
10313	10316	10319	10322	10325	10328	10331	10334	10337	10340	10343	10346	10349	10352	10355
10360	10363	10366	10369	10372	10375	10378	10381	10384	10387	10390	10393	10396	10399	10402
10440	10443	10446	10449	10452	10455	10458	10461	10464	10467	10470	10473	10476	10479	10482
10552	10555	10558	10561	10564	10567	10570	10573	10576	10579	10582	10585	10588	10591	10594
10600	10603	10606	10609	10612	10615	10618	10621	10624	10627	10630	10633	10636	10639	10642
10645	10648	10651	10654	10657	10660	10663	10666	10669	10672	10675	10678	10681	10684	10687
10707	10710	10713	10716	10719	10722	10725	10728	10731	10734	10737	10740	10743	10746	10749
10791	10794	10797	10800	10803	10806	10809	10812	10815	10818	10821	10824	10827	10830	10833
10857	10860	10863	10866	10869	10872	10875	10878	10881	10884	10887	10890	10893	10896	10899
10932	10935	10938	10941	10944	10947	10950	10953	10956	10959	10962	10965	10968	10971	10974
10980	10983	10986	10989	10992	10995	10998	11001	11004	11007	11010	11013	11016	11019	11022
11053	11056	11059	11062	11065	11068	11071	11074	11077	11080	11083	11086	11089	11092	11095
11121	11124	11127	11130	11133	11136	11139	11142	11145	11148	11151	11154	11157	11160	11163
11219	11222	11225	11228	11231	11234	11237	11240	11243	11246	11249	11252	11255	11258	11261
11264	11267	11270	11273	11276	11279	11282	11285	11288	11291	11294	11297	11300	11303	11306
11311	11314	11317	11320	11323	11326	11329	11332	11335	11338	11341	11344	11347	11350	11353
11391	11394	11397	11400	11403	11406	11409	11412	11415	11418	11421	11424	11427	11430	11433
11503	11506	11509	11512	11515	11518	11521	11524	11527	11530	11533	11536	11539	11542	11545
11551	11554	11557	11560	11563	11566	11569	11572	11575	11578	11581	11584	11587	11590	11593
11596	11599	11602	11605	11608	11611	11614	11617	11620	11623	11626	11629	11632	11635	11638
11658	11661	11664	11667	11670	11673	11676	11679	11682	11685	11688	11691	11694	11697	11700
11732	11735	11738	11741	11744	11747	11750	11753	11756	11759	11762	11765	11768	11771	11774
11838	11841	11844	11847	11850	11853	11856	11859	11862	11865	11868	11871	11874	11877	11880
11883	11886	11889	11892	11895	11898	11901	11904	11907	11910	11913	11916	11919	11922	11925
11928	11931	11934	11937	11940	11943	11946	11949	11952	11955	11958	11961	11964	11967	11970
11988	12004	12010	12013	12016	12019	12022	12025	12028	12031	12034	12037	12040	12043	12046
12075	12081	12086	12091	12096	12101	12106	12111	12116	12121	12126	12131	12136	12141	12146
12170	12173	12176	12179	12182	12185	12188	12191	12194	12197	12200	12203	12206	12209	12212
12215	12218	12221	12224	12227	12230	12233	12236	12239	12242	12245	12248	12251	12254	12257
12262	12265	12268	12271	12274	12277	12280	12283	12286	12289	12292	12295	12298	12301	12304
12312	12315	12318	12321	12324	12327	12330	12333	12336	12339	12342	12345	12348	12351	12354
12362	12365	12368	12371	12374	12377	12380	12383	12386	12389	12392	12395	12398	12401	12404
12412	12415	12418	12421	12424	12427	12430	12433	12436	12439	12442	12445	12448	12451	12454
12462	12465	12468	12471	12474	12477	12480	12483	12486	12489	12492	12495	12498	12501	12504
12512	12515	12518	12521	12524	12527	12530	12533	12536	12539	12542	12545	12548	12551	12554
12562	12565	12568	12571	12574	12577	12580	12583	12586	12589	12592	12595	12598	12601	12604
12612	12615	12618	12621	12624	12627	12630	12633	12636	12639	12642	12645	12648	12651	12654
12662	12665	12668	12671	12674	12677	12680	12683	12686	12689	12692	12695	12698	12701	12704
12712	12715	12718	12721	12724	12727	12730	12733	12736	12739	12742	12745	12748	12751	12754
12762	12765	12768	12771	12774	12777	12780	12783	12786	12789	12792	12795	12798	12801	12804
12812	12815	12818	12821	12824	12827	12830	12833	12836	12839	12842	12845	12848	12851	12854
12862	12865	12868	12871	12874	12877	12880	12883	12886	12889	12892	12895	12898	12901	12904
12912	12915	12918	12921	12924	12927	12930	12933	12936	12939	12942	12945	12948	12951	12954
12962	12965	12968	12971	12974	12977	12980	12983	12986	12989	12992	12995	12998	13001	13004
13012	13015	13018	13021	13024	13027	13030	13033	13036	13039	13042	13045	13048	13051	13054
13062	13065	13068	13071	13074	13077	13080	13083	13086	13089	13092	13095	13098	13101	13104
13112	13115	13118	13121	13124	13127	13130	13133	13136	13139	13142	13145	13148	13151	13154
13162	13165	13168	13171	13174	13177	13180	13183	13186	13189	13192	13195	13198	13201	13204
13212	13215	13218	13221	13224	13227	13230	13233	13236	13239	13242	13245	13248	13251	13254
13262	13265	13268	13271	13274	13277	13280	13283	13286	13289	13292	13295	13298	13301	13304
13312	13315	13318	13321	13324	13327	13330	13333	13336	13339	13342	13345	13348	13351	13354
13362	13365	13368	13371	13374	13377	13380	13383	13386	13389	13392	13395	13398	13401	13404
13412	13415	13418	13421	13424	13427	13430	13433	13436	13439	13442	13445	13448	13451	13454
13462	13465	13468	13471	13474	13477	13480	13483	13486	13489	13492	13495	13498	13501	13504
13512	13515	13518	13521	13524	13527	13530	13533	13536	13539	13542	13545	13548	13551	13554
13562	13565	13568	13571	13574	13577	13580	13583	13586	13589	13592	13595	13598	13601	13604
13612	13615	13618	13621	13624	13627	13630	13633	13636	13639	13642	13645	13648	13651	13654
13662	13665	13668	13671	13674	13677	13680	13683	13686	13689	13692	13695	13698	13701	13704
13712	13715	13718	13721	13724	13727	13730	13733	13736	13739	13742	13745	13748	13751	13754
13762	13765	13768	13771	13774	13777	13780	13783	13786	13789	13792	13795	13798	13801	13804
13812	13815	13818	13821	13824	13827	13830	13833	13836	13839	13842	13845	13848	13851	13854
13862	13865	13868	13871	13874	13877	13880	13883	13886	13889	13892	13895	13898	13901	13904
13912	13915	13918	13921	13924	13927	13930	13933	13936	13939	13942	13945	13948	13951	13954
13962	13965	13968	13971	13974	13977	13980	13983	13986	13989	13992	13995	13998	14001	14004
14012	14015	14018	14021	14024	14027	14030	14033	14036	14039	14042	14045	14048	14051	14054
14062	14065	14068	14071	14074	14077	14080	14083	14086	14089	14092	14095	14098	14101	14104
14112	14115	14118	14121	14124	14127	14130	14133	14136	14139	14142	14145	14148	14151	14154

CROSS REFERENCE TABLE -- MACRO NAMES

9716	9717	9718	9719	9720	9721	9722	9723	9724	9725	9726	9727	9728	9729	9730	9731	9732	9733	9734	9735	9736	9737	9738	9739	9740	9741	9742	9743	9744	9745	9746	9747	9748	9749	9750	9751	9752	9753	9754	9755	9756	9757	9758	9759	9760	9761	9762	9763	9764	9765	9766	9767	9768	9769	9770	9771	9772	9773	9774	9775	9776	9777	9778	9779	9780	9781	9782	9783	9784	9785	9786	9787	9788	9789	9790	9791	9792	9793	9794	9795	9796	9797	9798	9799	9800	9801	9802	9803	9804	9805	9806	9807	9808	9809	9810	9811	9812	9813	9814	9815	9816	9817	9818	9819	9820	9821	9822	9823	9824	9825	9826	9827	9828	9829	9830	9831	9832	9833	9834	9835	9836	9837	9838	9839	9840	9841	9842	9843	9844	9845	9846	9847	9848	9849	9850	9851	9852	9853	9854	9855	9856	9857	9858	9859	9860	9861	9862	9863	9864	9865	9866	9867	9868	9869	9870	9871	9872	9873	9874	9875	9876	9877	9878	9879	9880	9881	9882	9883	9884	9885	9886	9887	9888	9889	9890	9891	9892	9893	9894	9895	9896	9897	9898	9899	9900	9901	9902	9903	9904	9905	9906	9907	9908	9909	9910	9911	9912	9913	9914	9915	9916	9917	9918	9919	9920	9921	9922	9923	9924	9925	9926	9927	9928	9929	9930	9931	9932	9933	9934	9935	9936	9937	9938	9939	9940	9941	9942	9943	9944	9945	9946	9947	9948	9949	9950	9951	9952	9953	9954	9955	9956	9957	9958	9959	9960	9961	9962	9963	9964	9965	9966	9967	9968	9969	9970	9971	9972	9973	9974	9975	9976	9977	9978	9979	9980	9981	9982	9983	9984	9985	9986	9987	9988	9989	9990	9991	9992	9993	9994	9995	9996	9997	9998	9999
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

CROSS REFERENCE TABLE -- MACRO NAMES

9948	9951	9954	9957	9960	9963	9966	9969	9972	9975	9978	9981	9984	9987	9990
9993	9996	9999	10002	10005	10008	10011	10014	10017	10020	10023	10026	10029	10032	10035
10039	10043	10049	10052	10055	10058	10061	10064	10067	10070	10073	10076	10079	10082	10085
10108	10111	10114	10117	10120	10123	10126	10129	10132	10135	10138	10141	10144	10147	10150
10173	10179	10194	10229	10232	10235	10238	10241	10244	10247	10250	10253	10256	10259	10262
10268	10271	10274	10277	10280	10283	10286	10289	10292	10295	10298	10301	10304	10307	10310
10313	10316	10319	10322	10325	10328	10331	10334	10337	10340	10343	10346	10349	10352	10355
10360	10366	10369	10375	10378	10381	10384	10387	10390	10393	10396	10400	10406	10413	10419
10428	10431	10437	10440	10443	10446	10449	10452	10455	10458	10461	10467	10471	10477	10484
10496	10511	10546	10549	10552	10555	10558	10561	10564	10567	10570	10573	10576	10579	10582
10588	10591	10594	10597	10600	10603	10606	10609	10612	10615	10618	10621	10624	10627	10630
10633	10636	10639	10642	10645	10648	10651	10654	10657	10660	10663	10666	10669	10673	10677
10683	10686	10689	10692	10695	10698	10701	10704	10707	10710	10713	10716	10719	10722	10725
10748	10754	10757	10760	10763	10766	10769	10772	10775	10778	10781	10784	10787	10790	10793
10828	10863	10866	10869	10872	10875	10878	10881	10884	10887	10890	10893	10896	10899	10902
10908	10911	10914	10917	10920	10923	10926	10929	10932	10935	10938	10941	10944	10947	10950
10953	10956	10959	10962	10965	10968	10971	10974	10977	10980	10983	10986	10989	10992	10995
11003	11009	11015	11018	11021	11024	11027	11030	11033	11036	11039	11042	11045	11048	11051
11071	11074	11080	11083	11086	11089	11092	11095	11098	11101	11104	11107	11110	11113	11116
11180	11183	11186	11189	11192	11195	11198	11201	11204	11207	11210	11213	11216	11219	11222
11228	11231	11234	11237	11240	11243	11246	11249	11252	11255	11258	11261	11264	11267	11270
11273	11276	11279	11282	11285	11288	11291	11294	11297	11300	11303	11306	11309	11312	11315
11326	11329	11332	11335	11338	11341	11344	11347	11350	11353	11356	11359	11362	11365	11368
11391	11397	11403	11406	11409	11412	11415	11418	11421	11424	11427	11430	11433	11436	11439
11500	11503	11506	11509	11512	11515	11518	11521	11524	11527	11530	11533	11536	11539	11542
11548	11551	11554	11557	11560	11563	11566	11569	11572	11575	11578	11581	11584	11587	11590
11593	11596	11599	11602	11605	11608	11611	11614	11617	11620	11623	11626	11629	11632	11635
11649	11652	11655	11658	11661	11664	11667	11670	11673	11676	11679	11682	11685	11688	11691
11714	11720	11723	11726	11729	11732	11735	11738	11741	11744	11747	11750	11753	11756	11759
11820	11823	11826	11829	11832	11835	11838	11841	11844	11847	11850	11853	11856	11859	11862
11868	11871	11874	11877	11880	11883	11886	11889	11892	11895	11898	11901	11904	11907	11910
11913	11916	11919	11922	11925	11928	11931	11934	11937	11940	11943	11946	11949	11952	11955
11969	11972	11975	11978	11981	11984	11987	11990	11993	11996	12000	12003	12006	12009	12012
12037	12040	12043	12046	12049	12052	12055	12058	12061	12064	12067	12070	12073	12076	12079
12140	12143	12146	12149	12152	12155	12158	12161	12164	12167	12170	12173	12176	12179	12182
12188	12191	12194	12197	12200	12203	12206	12209	12212	12215	12218	12221	12224	12227	12230
12233	12236	12239	12242	12245	12248	12251	12254	12257	12260	12263	12266	12269	12272	12275
12289	12292	12295	12298	12301	12304	12307	12310	12313	12316	12319	12322	12325	12328	12331
12357	12360	12363	12366	12369	12372	12375	12378	12381	12384	12387	12390	12393	12396	12399
12460	12463	12466	12469	12472	12475	12478	12481	12484	12487	12490	12493	12496	12499	12502
12518	12511	12514	12517	12520	12523	12526	12529	12532	12535	12538	12541	12544	12547	12550
12593	12596	12599	12602	12605	12608	12611	12614	12617	12620	12623	12626	12629	12632	12635
12677	12680	12683	12686	12689	12692	12695	12698	12701	12704	12707	12710	12713	12716	12719
12780	12783	12786	12789	12792	12795	12798	12801	12804	12807	12810	12813	12816	12819	12822
12838	12831	12834	12837	12840	12843	12846	12849	12852	12855	12858	12861	12864	12867	12870
12897	12890	12893	12896	12899	12902	12905	12908	12911	12914	12917	12920	12923	12926	12929
12953	12956	12959	12962	12965	12968	12971	12974	12977	12980	12983	12986	12989	12992	12995
12997	13000	13003	13006	13009	13012	13015	13018	13021	13024	13027	13030	13033	13036	13039
13108	13106	13109	13112	13115	13118	13121	13124	13127	13130	13133	13136	13139	13142	13145
13148	13151	13154	13157	13160	13163	13166	13169	13172	13175	13178	13181	13184	13187	13190
13193	13196	13199	13202	13205	13208	13211	13214	13217	13220	13223	13226	13229	13232	13235
13249	13252	13255	13258	13261	13264	13267	13270	13273	13276	13279	13282	13285	13288	13291
13317	13320	13324	13330	13337	13343	13349	13354	13359	13364	13369	13374	13379	13384	13389
13423	13426	13429	13432	13435	13438	13441	13444	13447	13450	13453	13456	13459	13462	13465
13468	13471	13474	13477	13480	13483	13486	13489	13492	13495	13498	13501	13504	13507	13510

CROSS REFERENCE TABLE -- MACRO NAMES

13513	13516	13519	13522	13526	13530	13536	13539	13545	13551	13554	13557	13560	13563	13566
13570	13571	13583	13589	13595	13599	13601	13607	13610	13616	13622	13625	13630	13631	13634
13637	13641	13647	13654	13660	13666	13671	13674	13677	13682	13685	13686	13690	13691	13694
13742	13751	13751	13757	13767	13770	13774	13783	13785	13805	13808	13812	13823	13857	13860
13863	13866	13870	13873	13876	13879	13882	13886	13888	13892	13895	13900	13903	13906	13909
13912	13915	13918	13921	13927	13930	13933	13936	13939	13942	13945	13948	13951	13954	13957
13962	13971	13974	13980	13987	13996	14009	14021	14027	14031	14034	14046	14049	14053	14059
14063	14069	14073	14079	14083	14089	14093	14099	14103	14109	14113	14121	14125	14131	14135
14141	14145	14151	14155	14161	14165	14171	14175	14181	14185	14191	14195	14204	14207	14211
14217	14220	14223	14226	14230	14236	14239	14243	14249	14253	14256	14262	14265	14269	14275
14278	14282	14288	14291	14295	14301	14304	14308	14314	14317	14321	14327	14330	14334	14343
14346	14349	14352	14355	14361	14367	14370	14373	14376	14380	14408	14417	14420	14426	14432
14435	14438	14441	14444	14448	14481	14484	14487	14490	14493	14496	14499	14502	14505	14508
14511	14514	14517	14520	14523	14526	14529	14532	14535	14538	14541	14544	14550	14553	14556
14562	14565	14568	14574	14580	14583	14589	14595	14598	14604	14607	14613	14623	14626	14632
14638	14641	14644	14647	14650	14654	14654	14659	14702	14705	14708	14711	14714	14717	14720
14723	14726	14729	14732	14738	14747	14750	14753	14758	14784	14815	14818	14821	14824	14827
14830	14833	14836	14839	14843	14845	14851	14850	14853	14856	14859	14872	14891	14894	14897
14900	14903	14906	14909	14912	14915	14921	14930	14933	14936	14939	14942	14955	14958	14961
14964	14967	14970	14973	14976	14979	14985	14988	14991	14994	15010	15013	15016	15019	15022
15025	15028	15034	15041	15047										

SETUP
TRI

18	18	18	18	18	18	18	18	18	18	18	18	18	18	18
3330	2306	2386	2503	2583	2700	2780	2897	2977	3092	3104	3181	3193	3309	3324
4610	3407	3422	3428	3545	3625	3742	3822	3939	4019	4136	4216	4333	4413	4530
6129	4727	4807	4824	5004	5121	5191	5301	5386	5508	5593	5715	5800	5922	6007
7611	6214	6336	6421	6543	6628	6750	6835	6957	7042	7164	7249	7419	7456	7554
8733	7656	7696	7815	7861	8054	8176	8247	8333	8407	8483	8542	8573	8626	8702
9735	8792	9045	9107	9166	9178	9231	9412	9418	9424	9483	9485	9495	9548	9729
10375	9741	9800	9803	9813	9825	10046	10052	10058	10117	10120	10129	10182	10313	10369
11080	10434	10437	10446	10499	10680	10686	10692	10751	10754	10763	10816	10997	11019	11068
11767	11133	11314	11320	11326	11386	11388	11397	11450	11631	11637	11643	11702	11715	11714
12653	11948	11954	11960	12019	12023	12031	12064	12265	12277	12336	12348	12401	12582	12594
13533	12665	12718	12899	12911	12970	12982	13035	13216	13222	13228	13287	13290	13299	13352
14827	12812	12804	12816	12869	12980	14361	14366	14535	14541	14613	14632	14720	14729	14753

UNI

18	18	18	18	18	18	18	18	18	18	18	18	18	18	18
8235	5159	5162	5229	5232	7830	7952	7997	8003	8006	8009	8085	8091	8094	8097
8910	8238	8395	8398	8530	8533	8614	8617	8643	8696	8780	8783	8862	8865	8901
10884	8913	8916	8982	9233	9299	9550	9616	9867	9933	10184	10250	10501	10567	10818
12588	1003	11071	11135	11201	11452	11518	11769	11835	12086	12152	12271	12339	12403	12469
13965	12656	12720	12786	12905	12973	13037	13103	13354	13420	13539	13607	13671	13789	13792
	14003	14012	14037	14389	14398	14767	14770	14773	14809	14812	14885	14888		

.SACTI 18
 .SAPTB 18
 .SAPTH 18
 .SAPTY 18
 .SCATC 18
 .SCHTA 18
 .SEOP 18
 .SERRO 18
 .SERRT 18
 .SPOME 18
 .SROOC 18
 .SREAO 18
 .SROOP 18
 .STRAP 18
 .STYPE 18

217
295
481
1174
240
959
1662
1336
1233
1026
1521
1095

N13

DZKCA MACY11 27(1005) 13-MAY-77 14:07 PAGE 356

DZKCA.P11 13-MAY-77 13:58

CROSS REFERENCE TABLE -- MACRO NAMES

PAGE: 0376

.STYPO 18

.ABS. 037470 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DZKCA,DZKCA/SOL/CRF+DZKCA.MAK,DZKCA.MAC,DZKCA.P11/EQ:DZKCA

RUN-TIME: 219 224 13 SECONDS

RUN-TIME RATIO: 7717/457=16.8

CORE USED: SSK (110 PAGES)