

* JUKUBBSEQ
POP10 PAGE: 0001

00010000

770720

POP10 411

HDR1DZKCASEQ

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 the 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 8K 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 PROCEDURE

- 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

PC	CSR	STAT1	STAT2	STA13
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 MB201 TYPE "1", IF MB202 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

SW015	Set:	Halt on error
SW014	Set:	Loop on current test
SW013	Set:	Inhibit error print out
SW012	Set:	Inhibit type out abell on error.
SW011	Set:	Inhibit iterations. (quick pass)
SW010	Set:	Escape to next test on error
SW009	Set:	Loop with current data
SW008	Set:	Catch error and loop on it
SW007	Set:	Use previous status table.
SW006	Set:	Halt in ROMCLK routine before clocking micro-processor
SW005	Set:	Reserved
SW004	Set:	Reserved
SW003	Set:	Reselect KMC11's desired active
SW002	Set:	Lock on selected test
SW001	Set:	Restart program at selected test
SW000	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 KNC11'S DESIRED ACTIVE. Please note that a message is typed out for setting the switch register equal to KNC11's active. this means if the system has four KNC11s; 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 KNC11s are in the system *****DO NOT***** set switches greater than SW 03 in the up position. this would be a fatal error. do not select more active KNC11s 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 KNC desired active.
 EXAMPLE: If you have 4 KNC'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. SW 12 Delete print out/bell on error.
2. SW 13 Delete error printout.
3. SW 15 Halt on the error.
4. SW 08 Gate beginning of the test(on error).
5. SW 10 Gate 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 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
KMSTOO-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 MB201
BIT13=1 LINE UNIT IS AN MB202
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:

SWR=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 (SMR = 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.

*

MAINDEC-11-DZKCA-A

E02

DECDOC VER 00.04 13-MAY-77 16:16 PAGE 01 PAGE: 0017

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>

M3INDEC-11-DZYCA-A

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 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..

- 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 IBU, 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 MPMAM 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 XXX.
 1) CSR4:=GOOD DATA 2) CSR5:=BAD DATA
 3) CSR3:=ERROR TYPE 4) CSR6:=ERROR PC IN MICRO-CODE.
 5) CSR7:=MISCELLANEOUS INFORMATION.

M02

DZKCA MACY11 27(1006) 13-MAY-77 14:07 PAGE 3
DZKCA.P11 13-MAY-77 13:58

PAGE: 0025

1098
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

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

```

KHEBNT ↑?MAINDEC-11-DZKCA-A? DZKCA ↑?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-DZORC-C3), JAN 19, 1977.
.*
.HEADER <1976>,↑?MAINDEC-11-DZKCA-A?,<DZKCA>,<KMC11>,↑?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
.*SMR=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

.SBTL 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,PSM
STKLM= 177774 ;;STACK LIMIT REGISTER
PIRQ= 177772 ;;PROGRAM INTERRUPT REQUEST REGISTER
DSMR= 177570 ;;HARDWARE SWITCH REGISTER
DOISP= 177570 ;;HARDWARE DISPLAY REGISTER

.*GENERAL PURPOSE REGISTER DEFINITIONS
    
```


BASIC DEFINITIONS

```

142 001000 BIT09= 1000
143 000400 BIT08= 400
144 000200 BIT07= 200
145 000100 BIT06= 100
146 000040 BIT05= 40
147 000020 BIT04= 20
148 000010 BIT03= 10
149 000004 BIT02= 4
150 000002 BIT01= 2
151 000001 BIT00= 1
152 .EQUIV BIT09,BIT9
153 .EQUIV BIT08,BIT8
154 .EQUIV BIT07,BIT7
155 .EQUIV BIT06,BIT6
156 .EQUIV BIT05,BIT5
157 .EQUIV BIT04,BIT4
158 .EQUIV BIT03,BIT3
159 .EQUIV BIT02,BIT2
160 .EQUIV BIT01,BIT1
161 .EQUIV BIT00,BIT0
    
```

;;BASIC "CPU" TRAP VECTOR ADDRESSES

```

162 ERRTVEC= 4 ;; TIME OUT AND OTHER ERRORS
163 RESVVEC= 10 ;; RESERVED AND ILLEGAL INSTRUCTIONS
164 TRITVEC= 14 ;; T BIT
165 TRIVVEC= 14 ;; TRACE TRAP
166 BPTVEC= 14 ;; BREAKPOINT TRAP (BPT)
167 IOTVEC= 20 ;; INPUT/OUTPUT TRAP (IOT) **SCOPE**
168 PWRVEC= 20 ;; POWER FAIL
169 ENTVEC= 30 ;; EMULATOR TRAP (ENT) **ERROR**
170 TRAPVEC= 34 ;; TRAP TRAP
171 TKVEC= 60 ;; TTY KEYBOARD VECTOR
172 TPVEC= 64 ;; TTY PRINTER VECTOR
173 PIRGVEC= 240 ;; PROGRAM INTERRUPT REQUEST VECTOR
    
```

;; INSTRUCTION DEFINITIONS

```

181
182
183 PUSH1SP=5746 ;; DECREMENT PROCESSOR STACK 1 WORD
184 POP1SP=5726 ;; INCREMENT PROCESSOR STACK 1 WORD
185 PUSHAD=10046 ;; SAVE AD ON STACK
186 POPAD=12600 ;; RESTORE AD FROM STACK
187 PUSH2SP=24646 ;; DECREMENT STACK TWICE
188 POP2SP=22626 ;; INCREMENT STACK TWICE
189 .EQUIV ENT,HLT ;; BASIC DEFINITION OF ERROR CALL
190
191
192
    
```

TRAPCATCHER FOR UNEXPECTED INTERRUPTS

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
(2)
(2)
238
239

; TRAPCATCHER FOR ILLEGAL INTERRUPTS
; THE STANDARD "TRAP CATCHER" IS PLACED
; BETWEEN ADDRESS 0 TO ADDRESS 776.
; IT LOOKS LIKE "PC+2 HALT".

. = 0
; .WORD 0,0
; STANDARD INTERRUPT VECTORS

. = 20
; SSCOPE : SCOPE LOOP HANDLER.
; PR7 : SERVICE AT LEVEL 7.
; SPMON : POWER FAIL HANDLER
; PR7 : SERVICE AT LEVEL 7
; SERROR : ERROR HANDLER
; PR7 : SERVICE AT LEVEL 7
; STRAP : GENERAL HANDLER DISPATCH SERVICE
; PR7 : SERVICE AT LEVEL 7
; SBTTL ACT11 HOOKS

; HOOKS REQUIRED BY ACT11

. = 46
; SSVPC = . ; SAVE PC
; SENDAD : 1) SET LOC. 46 TO ADDRESS OF SENDAD IN .SEOP
. = 52
; .WORD 0 ; 2) SET LOC. 52 TO ZERO
; . = SSVPC ; RESTORE PC

. = 174
DISPREG: 0 ; SOFTWARE DISPLAY REGISTER
SWREG: 0 ; SOFTWARE SWITCH REGISTER

. = 200
JMP .START ; GO TO START OF PROGRAM

. = 1000
MTITLE: .ASCII <200><12>/MAINDEC-11-DZKCA-A/<200>
.ASCII /BASIC KMC11 CONTROLLER TEST/<200>

DSMR = 177570
DDISP = 177570

298
299
300 001316 000000
301 001320 000000
302 001322 000000
303 001324 000000
304 001326 000000
305 001330 000000
306 001332 000000
307 001334 000000
308 001336 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
329 001352 000
330 001354 000
331 001356 000
332 001358 000
333 001360 000
334 001362 000
335 001364 000
336 001366 000
337 001368 000
338 001370 000
339 001372 000
340 001374 000
341 001376 000
342 001378 000
343 001380 000
344 001382 000
345 001384 000
346 001386 000
347 001388 000
348 001390 000
349 001392 000
350 001394 000
351 001396 000

```

*****
: .EVEN
: SMAIL: .WORD AMSGTY :; APT MAILBOX
: SMSGTY: .WORD AFATAL :; MESSAGE TYPE CODE
: SFATAL: .WORD ATESTN :; FATAL ERROR NUMBER
: STESTN: .WORD APAS :; TEST NUMBER
: SPASS: .WORD ADEV :; TEST COUNT
: SDEVCT: .WORD AUNT :; DEVICE COUNT
: SUNIT: .WORD :; 70 UNIT NUMBER
: SMSGNO: .WORD :; ADDRESS
: SMSGG: .WORD :; LENGTH
: SETABLE :; INCREMENT TABLE
: SENV: .BYTE AENV :; ENVIRONMENT BYTE
: SENVM: .BYTE AENVN :; ENVIRONMENT MODE BITS
: SSWREG: .WORD ASWREG :; APT SWITCH REGISTER
: SUSWR: .WORD AUSWR :; USER SWITCHES
: SCPUOP: .WORD ACPUOP :; CPU TYPE, OPTIONS
: * :; BITS 15-11=CPU TYPE
: * :; 11/0=01,11/05=02,11/20=03,11/40=04,11/45=05
: * :; 11/70=06,PD=07,0=10
: * :; BIT 10=REAL TIME CLOCK
: * :; BIT 9=FLOATING POINT PROCESSOR
: * :; BIT 8=MEMORY MANAGEMENT
: SHMS1: .BYTE AHMS1 :; HIGH ADDRESS, H.S. BYTE
: SHY1: .BYTE AHY1 :; MEM. TYPE, BLK#1
: * :; MEM. TYPE BYTE -- (HIGH BYTE)
: * :; 500 MSEC CORE=001
: * :; 300 MSEC BIPOLAR=002
: * :; 500 MSEC MOS=003
: SHADR1: .WORD AHADR1 :; HIGH ADDRESS, BLK#1
: * :; MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
: SHMS2: .BYTE AHMS2 :; HIGH ADDRESS, H.S. BYTE
: SHY2: .BYTE AHY2 :; MEM. TYPE, BLK#2
: SHADR2: .WORD AHADR2 :; MEM. LAST ADDRESS, BLK#2
: SHMS3: .BYTE AHMS3 :; HIGH ADDRESS, H.S. BYTE
: SHY3: .BYTE AHY3 :; MEM. TYPE, BLK#3
: SHADR3: .WORD AHADR3 :; MEM. LAST ADDRESS, BLK#3
: SHMS4: .BYTE AHMS4 :; HIGH ADDRESS, H.S. BYTE
: SHY4: .BYTE AHY4 :; MEM. TYPE, BLK#4
: SHADR4: .WORD AHADR4 :; MEM. LAST ADDRESS, BLK#4
: SVECT1: .WORD AVECT1 :; INTERRUPT VECTOR#1, BUS PRIORITY#1
: SVECT2: .WORD AVECT2 :; INTERRUPT VECTOR#2, BUS PRIORITY#2
: SBASE: .WORD :; BASE ADDRESS OF EQUIPMENT UNDER TEST
: SDEVN: .WORD ADEVN :; DEVICE #
: SCOM1: .WORD ACOM1 :; CONTROLLER DESCRIPTION WORD#1
: SCOM2: .WORD ACOM2 :; CONTROLLER DESCRIPTION WORD#2
: SDDM0: .WORD ADDM0 :; DEVICE DESCRIPTOR WORD#0
: SDDM1: .WORD ADDM1 :; DEVICE DESCRIPTOR WORD#1
: SDDM2: .WORD ADDM2 :; DEVICE DESCRIPTOR WORD#2
: SDDM3: .WORD ADDM3 :; DEVICE DESCRIPTOR WORD#3
: SDDM4: .WORD ADDM4 :; DEVICE DESCRIPTOR WORD#4
: SDDM5: .WORD ADDM5 :; DEVICE DESCRIPTOR WORD#5
: SDDM6: .WORD ADDM6 :; DEVICE DESCRIPTOR WORD#6
: SDDM7: .WORD ADDM7 :; DEVICE DESCRIPTOR WORD#7
: SDDM8: .WORD ADDM8 :; 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 001442
 361
 362
 363
 364
 365 001442 000000
 366 001444 000000
 367
 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 000
 393 001511 000
 394
 395
 396

SDDW9: .WORD ADDW9 ;; DEVICE DESCRIPTOR WORD#9
 SDDW10: .WORD ADDW10 ;; DEVICE DESCRIPTOR WORD#10
 SDDW11: .WORD ADDW11 ;; DEVICE DESCRIPTOR WORD#11
 SDDW12: .WORD ADDW12 ;; DEVICE DESCRIPTOR WORD#12
 SDDW13: .WORD ADDW13 ;; DEVICE DESCRIPTOR WORD#13
 SDDW14: .WORD ADDW14 ;; DEVICE DESCRIPTOR WORD#14
 SDDW15: .WORD ADDW15 ;; 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

STRTSW: .WORD 0 ; SWITCHES AT START OF PROGRAM
 STAT: .WORD 0 ; KM STATUS WORD STORAGE
 CLKX: .WORD 0
 MASIX: .WORD 0
 SAVSP: .WORD 0 ; STACK POINTER STORAGE
 SAVPC: .WORD 0 ; PROGRAM COUNTER STORAGE
 ZERO: .WORD 0
 ONE: .WORD 1
 NEPLIN: .WORD 0 ; HIGHEST LOCATION FOR NPR'S
 KMACTV: .BLKW 1 ; KMC11 SELECTED ACTIVE
 KANUM: .BLKW 1 ; OCTAL NUMBER OF KMC11'S
 SAVACT: .BLKW 1 ; ORIGINAL ACTIVE DEVICES.
 SAVNUM: .BLKW 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 KMC11 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 (SERRTB).
;*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

0
0
0
EM1 ;: ERROR 1
DH1
DT1
EM2 ;: ERROR 2
DH1
DT1
EM3 ;: ERROR 3
DH2
DT2
EM4 ;: ERROR 4
DH1
DT1
EM5 ;: ERROR 5
DH1
DT1
EM6 ;: ERROR 6
DH3
DT1
EM7 ;: ERROR 7
DH3
DT1
EM10 ;: ERROR 10
DH3
DT1
EM11 ;: ERROR 11
DH2
DT2
EM12 ;: ERROR 12
DH2
DT2
EM13 ;: ERROR 13
0
0
EM14 ;: ERROR 14
0

453	001630	036666	EM15	
454	001632	037373	DH4	; 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	DH2	; 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	DH4	; 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	DH2	; ERROR 25
479	001714	037456	DT2	
480		002034		

.=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-PDP11 DIAGNOSTIC
INTERFACE SPEC.

```

481			SAPTHD:	
482			SHIBTS:	.WORD 0 ;; TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
483			SMBADR:	.WORD SMBAL ;; ADDRESS OF APT MAILBOX (BITS 0-15)
484			STSTM:	.WORD 90. ;; RUN TIM OF LONGEST TEST
485			SPRSTM:	.WORD 95. ;; RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
486	002034	000000	SUNITH:	.WORD 95. ;; ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
487	000024	000200		.WORD SETEND-SMIAL/2 ;; LENGTH MAILBOX-ETABLE(WORDS)
488	000044	000044		
489		002034		
490		002034		
491		002034		
492				
493				
494				
495				
496	002034			
497	002034	000000		
498	002036	001316		
499	002040	000132		
500	002042	000137		
501	002044	000137		
502	002046	000052		
503				

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

002050 000000
002052 000000
002054 000000

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

;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
KMCSSR: 0 ; POINTER TO KMC11 CONTROL STATUS REGISTER
KMCSSRH: 0 ; POINTER TO KMC11 CONTROL STATUS REGISTER HIGH BYTE.
KMCCTL: 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

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

.=2100
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

565	002150	000001	KMCR05: .BLKW	1	:CONTROL STATUS REGISTER FOR KMC11 NUMBER 05
566	002152	000001	KMS105: .BLKW	1	:VECTOR FOR KMC11 NUMBER 05
567	002154	000001	KMS205: .BLKW	1	:DDCMP LINE# FOR KMC11 NUMBER 05
568	002156	000001	KMS305: .BLKW	1	:3RD STATUS WORD
569	002160	000001	KMCR06: .BLKW	1	:CONTROL STATUS REGISTER FOR KMC11 NUMBER 06
570	002162	000001	KMS106: .BLKW	1	:VECTOR FOR KMC11 NUMBER 06
571	002164	000001	KMS206: .BLKW	1	:DDCMP LINE# FOR KMC11 NUMBER 06
572	002166	000001	KMS306: .BLKW	1	:3RD STATUS WORD
573	002170	000001	KMCR07: .BLKW	1	:CONTROL STATUS REGISTER FOR KMC11 NUMBER 07
574	002172	000001	KMS107: .BLKW	1	:VECTOR FOR KMC11 NUMBER 07
575	002174	000001	KMS207: .BLKW	1	:DDCMP LINE# FOR KMC11 NUMBER 07
576	002176	000001	KMS307: .BLKW	1	:3RD STATUS WORD
577	002200	000001	KMCR10: .BLKW	1	:CONTROL STATUS REGISTER FOR KMC11 NUMBER 10
578	002202	000001	KMS110: .BLKW	1	:VECTOR FOR KMC11 NUMBER 10
579	002204	000001	KMS210: .BLKW	1	:DDCMP LINE# FOR KMC11 NUMBER 10
580	002206	000001	KMS310: .BLKW	1	:3RD STATUS WORD
581	002210	000001	KMCR11: .BLKW	1	:CONTROL STATUS REGISTER FOR KMC11 NUMBER 11
582	002212	000001	KMS111: .BLKW	1	:VECTOR FOR KMC11 NUMBER 11
583	002214	000001	KMS211: .BLKW	1	:DDCMP LINE# FOR KMC11 NUMBER 11
584	002216	000001	KMS311: .BLKW	1	:3RD STATUS WORD
585	002220	000001	KMCR12: .BLKW	1	:CONTROL STATUS REGISTER FOR KMC11 NUMBER 12
586	002222	000001	KMS112: .BLKW	1	:VECTOR FOR KMC11 NUMBER 12
587	002224	000001	KMS212: .BLKW	1	:DDCMP LINE# FOR KMC11 NUMBER 12
588	002226	000001	KMS312: .BLKW	1	:3RD STATUS WORD
589	002230	000001	KMCR13: .BLKW	1	:CONTROL STATUS REGISTER FOR KMC11 NUMBER 13
590	002232	000001	KMS113: .BLKW	1	:VECTOR FOR KMC11 NUMBER 13
591	002234	000001	KMS213: .BLKW	1	:DDCMP LINE# FOR KMC11 NUMBER 13
592	002236	000001	KMS313: .BLKW	1	:3RD STATUS WORD
593	002240	000001	KMCR14: .BLKW	1	:CONTROL STATUS REGISTER FOR KMC11 NUMBER 14
594	002242	000001	KMS114: .BLKW	1	:VECTOR FOR KMC11 NUMBER 14
595	002244	000001	KMS214: .BLKW	1	:DDCMP LINE# FOR KMC11 NUMBER 14
596	002246	000001	KMS314: .BLKW	1	:3RD STATUS WORD
597	002250	000001	KMCR15: .BLKW	1	:CONTROL STATUS REGISTER FOR KMC11 NUMBER 15
598	002252	000001	KMS115: .BLKW	1	:VECTOR FOR KMC11 NUMBER 15
599	002254	000001	KMS215: .BLKW	1	:DDCMP LINE# FOR KMC11 NUMBER 15
600	002256	000001	KMS315: .BLKW	1	:3RD STATUS WORD
601	002260	000001	KMCR16: .BLKW	1	:CONTROL STATUS REGISTER FOR KMC11 NUMBER 16
602	002262	000001	KMS116: .BLKW	1	:VECTOR FOR KMC11 NUMBER 16
603	002264	000001	KMS216: .BLKW	1	:DDCMP LINE# FOR KMC11 NUMBER 16
604	002266	000001	KMS316: .BLKW	1	:3RD STATUS WORD
605	002270	000001	KMCR17: .BLKW	1	:CONTROL STATUS REGISTER FOR KMC11 NUMBER 17
606	002272	000001	KMS117: .BLKW	1	:VECTOR FOR KMC11 NUMBER 17
607	002274	000001	KMS217: .BLKW	1	:DDCMP LINE# FOR KMC11 NUMBER 17
608	002276	000001	KMS317: .BLKW	1	:3RD STATUS WORD
609					
610					
611					
612					
613					
614					
615					

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

Code	Value	Label	Description
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	
CSR	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
STAT2	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	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 (DDCMP 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:
002410 012706 001200
002414 012737 007100 000024
002418 012737 001472 001476
002422 005137 011422
002426 105127 001203
002430 105127 001511
002434 012737 002070 001502
002438 012737 002276 001504
002442 012737 100000 001500
002446 012700 002302
002450 005020
002454 022700 002402 235:
002458 001374
002502 005137 001216
002506 012737 000001
002514 012737 002402 0

PROGRAM INITIALIZATION
LOCK OUT INTERRUPTS
SET UP PROCESSOR STACK
SET UP POWER FAIL VECTOR
CLEAR PROGRAM CONTROL FLAGS AND COUNTS
TYPE TITLE MESSAGE
MOV 8340,PS
MOV 8STACK,SP
MOV 8SPADR,2824
MOV 8KUNUM,8AVNUM
CLR 8WFLG
CLR 8ERRFLG
CLR 8OV.FLG
MOV 8CNT.MAP-10,CREAM
MOV 8CNT.MAP-4,MILK
MOV 8BIT15,RUN
MOV 8CNT.MAP,RO
CLR (RO)+
CMP 8CNT.MAP+100,RO
BNE 235
CLR 8ERRPC
MOV 81,8STSTM
MOV 8.START,8LPAOR
BIT 81,8ENV
BZ 35
MOV 8SMREG,8MREG
BR 65+2
MOV 286,-(SP)
MOV 284,-(SP)
MOV 865,284
MOV 8177570,8SMR
MOV 8177570,8DISPLAY
CMP 8-1,28SMR
BEQ 65+2
BR 75
CMP (SP)+,(SP)+
MOV 8SUREG,8SMR
MOV 8DISPREG,8DISPLAY
75:
MOV (SP)+,284
MOV (SP)+,286
TSTB 8INIFLG
BNE 205
CMP 8SENDAD,2842
BEQ 205
TYPE .MTITLE
JSR PC,8CKSMR
MOV 28SMR,8STRTSM
TST 2842
BEQ .+6
CLR 8STRTSM
BIT 8SM00,8STRTSM
BNE 175
TSTB 8STRTSM

LOCK OUT INTERRUPTS
SET UP STACK
SET UP POWER FAIL VECTOR
SAVE NUMBER OF DEVICES IN SYSTEM.
CLEAR SOFT TIMEOUT FLAG
CLEAR ERROR FLAG
ZERO QUICK VERIFY FLAG
GET MAP POINTER.
GET PASS COUNT MAP POINTER
POINT POINTER TO FIRST DEVICE.
PASS COUNT POINTER TO RO
CLEAR TABLE
DONE YET?
KEEP GOING
CLEAR LAST ERROR POINTER
SET UP FOR TEST 1
SET UP FOR POWER FAIL BEFORE TESTING STARTS
IS IT RUNNING UNDER APT?
IF NOT CHECK FOR TYPE OF SWITCH REGISTER.
LOAD SOFTWARE SWITCH REG.
GO SET UP SOFTWARE SWITCH REG.
SAVE CURRENT VECTORS
SET UP FOR TIMEOUT
SET SMR TO HARD SMR ADDRESS
SET DISPLAY TO HARD SMR ADDRESS
REFERENCE HARDWARE SWITCH REGISTER
IF = -1 USE SOFT SMR ANYWAY
IF IT EXISTS AND NOT = -1 USE HARD SMR
ADJUST STACK
POINTER TO SOFT SMR
POINTER TO SOFT DISPLAY REG
RESTORE VECTORS
HAS INITIALIZATION BEEN PERFORMED
BR IF YES
IF ACT-11 AUTOMATIC MODE, DON'T TYPE ID
TYPE TITLE MESSAGE
CHECK FOR SOFT SMR
STORE STARTING SWITCHES
IS IT RUNNING IN AUTO MODE?
BR IF NO
IF YES, CLEAR SWITCHES
IF SM00=1, QUESTIONS ARE ASKED.
BR IF SM00=1
BIT7=1??

PROGRAM INITIALIZATION AND START UP.

836	003146	012702	003342	4S:	MOV	#DEVTAB,R2	:R2 IS DEVICE TABLE PONTER
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	003163	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,#4	:OK ONLY KMC'S ARE LEFT, SET UP FOR TIMEOUT
846	003202	005711		FY:	TST	(R1)	:CHECK KMC ADDRESS
847	003204	020137	002066		CMF	R1,KMCSR	:DOES IT MATCH
848	003210	001403			BEQ	OK	:BR IF YES
849	003213	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:	CMFB	(R2)+,-(R3)	:ON TIMEOUT, INC R2, DEC R3
857	003242	000002			RTI		:SLPAOR
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			CMVRT		:TYPE OUT ERROR PC
864	003266	003322			ERRPC		
865	003270	104401			TYPE		:TYPE REST OF HEADER
866	003272	011001			CNERR		
867	003274	012737	177777	001302	MOV	#-1,\$TMP2	:SET FLAG SO IT ONLY GETS TYPED ONCE
868	003302	010137	001264	1S:	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			KMCM		
873	003316	022636		4S:	CMF	(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	:DG
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

F04

MAY 11 27(1006) 13-MAY-77 14:07 PAGE 22
13-MAY-77 13:58

PROGRAM INITIALIZATION AND START UP.

```

748 003634 012737 011464 001206 3$:  MOV  @CYCLE,$LPADR ; START AT "CYCLE" FIND WHICH DEVICE TO TEST
749 003642 032737 000002 001446 4$:  BIT  @SW01,STRTSW ; IS TEST NO. SELECTED?
950 003650 001002                BNE  5$ ; BR IF YES
951 003652 104401 007614                TYPE MR ; TYPE R
952 003656 000177 175324                5$:  JMP  @SLPADR ; START TESTING

```

: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 THERE'S A MONITOR GO TO IT
:IF THERE ISN'T JUMP TO CYCLE

SEOP:

003662
003662
003664
003670
970 003674
971 003700
972 003704
973 003710
974 003714
975 003720
976 003724
977 003730
978 003734
979 003740
980 003744
981 003750
982 003754
983 003762
984 003766
985 003774
986 004000
987 004004
988 004006
989 004014
990 004022
991 004026
992 004028
993 004028
994 004044
995 004046
996 004050
997 004052
998 004054
999 004056
1000 004060
1001 004064
1002 004066
1003 004070
1004 004072
1005 004074
1006 004076
1007 004100
1008 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
004050
013700 000042
001405
000005
004710
000240
000240
000240
000137

RESET
INC SPASS
CLRB SERFLG
TYPE ,NEPASS
TYPE ,PCSRX
CNVRT ,XCSR
TYPE ,MVECX
CNVRT ,XVEC
TYPE ,MPASSX
CNVRT ,XPASS
TYPE ,MERRX
CNVRT ,XERR
MOV RILK,RO
MOV SPASS,(RO)+
MOV SBTTL,(RO)+
MOV KRLVL,20RVEC
CLR 20RVL
MOV KRTLVL,20TVEC
CLR 20TVL
DEC SAVNUM
SDOAGN
BNE
MOVB 2377,OV.FLG
MOV KRNUM,SAVNUM
CLR SERPC
CLR STINES
INC SPASS
BIC 810000,SPASS
DEC (PC)+
SEOPCT: .WORD 1
BGT SDOAGN
MOV (PC)+,2(PC)+
SENDCT: .WORD 1
SEOPCT
\$GET42: MOV 2#42,RO
BEQ SDOAGN
RESET
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	

```

SRTMAD: .WORD   CYCLE
XCSR:   1
        .BYTE   6,2
        KNCSR
XVEC:   1
        .BYTE   4,2
        KMRVEC
XPASS:  1
        .BYTE   6,2
        SPASS
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(SISTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
*AND LOAD THE ERROR FLAG (SERFLG) 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

$SCOPE:  CLR      SERRPC          ; CLEAR LAST ERROR PC
          CMP      TST1+2,(SP)    ; IS THIS TEST #1 ?
          BEQ     $XTSTR          ; IF SO DON'T LOOP.
          BR      1$              ;
          TSTB   @STKS            ; KEYBOARD DONE ?
          BPL     $OVER           ; IF NO DONT WAIT.
          MOV     @STKB,-2(SP)
          BIT     @BIT14,@SWR      ; LOOP ON PRESENT TEST?
          BNE     $OVER           ; YES IF SM14=1
          ;*****START OF CODE FOR THE XOR TESTER*****
          $XTSTR: BR      6$
          MOV     @ERRVEC,-(SP)    ; IF RUNNING ON THE "XOR" TESTER CHANGE
          MOV     @SS,@ERRVEC     ; THIS INSTRUCTION TO A "NOP" (NOP=240)
          TST     @177060         ; SAVE THE CONTENTS OF THE ERROR VECTOR
          BR      @ERRVEC        ; SET FOR TIMEOUT
          BR      $SVLAD         ; TIME OUT ON XOR?
          CMP     (SP)+,(SP)+     ; RESTORE THE ERROR VECTOR
          MOV     (SP)+,@ERRVEC   ; GO TO THE NEXT TEST
          BR      $OVER         ; CLEAR THE STACK AFTER A TIME OUT
          ;*****END OF CODE FOR THE XOR TESTER*****
          TSTB   SERFLG          ; HAS AN ERROR OCCURRED?
          BEQ     3$              ; BR IF NO
          CLRB   SERFLG          ; ZERO THE ERROR FLAG
          CLR    STINES           ; CLEAR THE NUMBER OF ITERATIONS TO MAKE
          BIT    @BIT11,@SWR     ; 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      SICNT      ;: INCREMENT ITERATION COUNT
1069 004272 023737 001310 001204  CMP      STIMES,SICNT ;: CHECK THE NUMBER OF ITERATIONS MADE
1070 004300 002015      BGE      SOVER      ;: BR IF MORE ITERATION REQUIRED
1071 004302 012737 000001 001204 1S:  MOV     R1,SICNT    ;: REINITIALIZE THE ITERATION COUNTER
1072 004310 013737 004356 001310  MOV     SMAXCNT,STIMES ;: SET NUMBER OF ITERATIONS TO DO
1073 004316 105237 001202  SSVLAD: INCB     STSTN     ;: COUNT TEST NUMBERS
1074 004322 113737 001202 001322  MOV     STSTN,STSTN   ;: SET TEST NUMBER IN APT MAILBOX
1075 004330 011637 001206      MOV     (SP),SLPADR   ;: SAVE SCOPE LOOP ADDRESS
1076 004334 013777 001202 174700 SOVER:  MOV     STSTN,DISPLAY ;: DISPLAY TEST NUMBER
1077 004342 013716 001206      MOV     SLPADR,(SP)  ;: FUDGE RETURN ADDRESS
1078 004346 013701 002066      MOV     KMCSR,R1     ;: R1 CONTAINS BASE KMC ADDRESS.
1079 004352 000002      RTI
1080 004354 004406      BRW:    WORD      406
1081 004356 000020      SMAXCNT: 20          ;:MAX. NUMBER OF ITERATIONS
1082
1083      ;CHECK FOR FREEZE ON CURRENT DATA
1084      -----
1085
1086 004360 004737 011216 .SCOPI: JSR     PC,CKSWR     ;:CHECK FOR SOFT SWR
1087 004364 032777 001000 174646  BIT     #SM09,SMR     ;: 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:      SFILL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
1101 ;: #NOTE2:      SFILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
1102 ;: #NOTE3:      SFILLC 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  STYPE:  TSTB     STPFLG   ;: 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      3S          ;: LEAVE
1116 004416 010046 1S:     MOV     R0,-(SP)  ;: SAVE R0
1117 004420 017500 000002      MOV     @2(SP),R0    ;: GET ADDRESS OF ASCIZ STRING
1118 004424 122737 000001 001336  CMPB   #APTENV,SENV   ;: RUNNING IN APT MODE
1119 004432 001011      BNE     62S         ;: NO, GO CHECK FOR APT CONSOLE
1120 004434 132737 000100 001337  BITB   #APTSPool,SENYM ;: SPOOL MESSAGE TO APT

```

```

1121 004442 001405 BEQ 625 ; NO GO CHECK FOR CONSOLE
1122 004444 010037 004454 MOV 60,615 ; SETUP MESSAGE ADDRESS FOR APT
1123 004446 004737 004674 JSR PC,6ATY3 ; SPOOL MESSAGE TO APT
1124 004448 000000 ; MESSAGE ADDRESS
1125 004448 000000 61S: .WORD 0 ; APT CONSOLE SUPPRESSED
1126 004448 132737 000040 001337 62S: BITB 8APTCSUP,SENVM ; YES SKIP TYPE OUT
1127 004448 001000 BNE 60S ; PUSH CHARACTER TO BE TYPED ONTO STACK
1128 004448 112046 2S: MOVB (R0)+,-(SP) ; BR IF IT ISN'T THE TERMINATOR
1129 004470 001006 BNE 4S ; IF TERMINATOR POP IT OFF THE STACK
1130 004472 006726 TST (SP)+ ; RESTORE R0
1131 004472 018600 000002 60S: MOV (SP)+,R0 ; ADJUST RETURN PC
1132 004472 006716 3S: ADD #2,(SP) ; RETURN
1133 004604 122716 000011 4S: CMPB #HT,(SP) ; BRANCH IF <HT>
1134 004610 001430 BEQ 8S ; BRANCH IF NOT <CRLF>
1135 004612 122716 000200 CMPB #CRLF,(SP) ; POP <CR><LF> EQUIV
1136 004616 001006 BNE 5S ; TYPE A CR AND LF
1137 004620 006726 TST (SP)+ ; CLEAR CHARACTER COUNT
1138 004620 104401 TYPE ; GET NEXT CHARACTER
1139 004620 001313 SCLF ; GO TYPE THIS CHARACTER
1140 004626 105037 004662 CLRB #SCHARCNT ; IS IT TIME FOR FILLER CHARS.?
1141 004632 000755 BR 2S ; IF NO GO GET NEXT CHAR.
1142 004634 004737 004616 5S: JSR PC,STYPEC ; GET # OF FILLER CHARS. NEEDED
1143 004640 123726 001256 6S: CMPB #FILLC,(SP)+ ; AND THE NULL CHAR.
1144 004642 001360 BNE 6S ; DOES A NULL NEED TO BE TYPED?
1145 004646 013746 001254 MOV #NULL,-(SP) ; BR IF NO--GO POP THE NULL OFF OF STACK
1146 ; GO TYPE A NULL
1147 004652 105366 000001 7S: DECB 1,(SP) ; DO NOT COUNT AS A COUNT
1148 004656 002770 BLT 6S ; LOOP
1149 004660 004737 004616 JSR PC,STYPEC
1150 004664 105337 004662 DECB #SCHARCNT
1151 004670 000770 BR 7S
1152 ;HORIZONTAL TAB PROCESSOR
1153
1154
1155 004572 112716 000040 8S: MOVB #' '(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 9S ; TAB STOP
1159 004612 005726 TST (SP)+ ; POP SPACE OFF STACK
1160 004614 000724 BR 2S ; GET NEXT CHARACTER
1161 004616 105777 174426 $TYPEC: STTB #STPS ; WAIT UNTIL PRINTER IS READY
1162 004622 100375 BPL $TYPEC
1163 004624 116677 000002 174420 MOVB 2(SP),#STPB ; LOAD CHAR TO BE TYPED INTO DATA REG.
1164 004632 122766 000015 000002 CMPB #CR,2(SP) ; IS CHARACTER A CARRIAGE RETURN?
1165 004640 001003 BNE 1S ; BRANCH IF NO
1166 004642 105037 004662 CLRB #SCHARCNT ; YES--CLEAR CHARACTER COUNT
1167 004646 000406 BR $TYPEX ; EXIT
1168 004650 122766 000012 000002 1S: CMPB #LF,2(SP) ; IS CHARACTER A LINE FEED?
1169 004656 001402 BEQ $TYPEX ; BRANCH IF YES
1170 004660 105227 INCB (PC)+ ; COUNT THE CHARACTER
1171 004662 000000 $SCHARCNT: .WORD 0 ; CHARACTER COUNT STORAGE
1172 004664 000207 $TYPEX: RTS PC
1173
1174 .SBTTL APT COMMUNICATIONS ROUTINE
1175
1176 ;*****

```

APT COMMUNICATIONS ROUTINE

```

1177 004656 112737 000001 005132 SATY1: MOVB #1, SFFLG ;; TO REPORT FATAL ERROR
1178 004659 112737 000001 005130 SATY3: MOVB #1, SFFLG ;; TO TYPE A MESSAGE
1179 004702 000403 BR SATYC
1180 004704 112737 000001 005132 SATY2: 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 CFPB APTENV, SENV ;; OPERATING UNDER APT?
1187 004732 001031 BNE 35 ;; IF NOT: BR
1188 004734 132737 000100 001337 BITB APTSPool, 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, SMSGAD ;; PUT ADDR IN MAILBOX
1195 004770 105720 25: TSTB (RO)+ ;; FIND END OF MESSAGE
1196 004772 001376 BNE 25
1197 004774 163700 001332 SUB SMSGAD, RO ;; SUB START OF MESSAGE
1198 005000 006200 ASR RO ;; GET MESSAGE LENGTH 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 005050 001416 BEQ 125 ;; IF NOT: BR
1210 005052 005737 001336 TST SENV ;; RUNNING UNDER APT?
1211 005056 001413 BEQ 125 ;; IF NOT: BR
1212 005060 005737 001316 115: TST SMSGTYPE ;; FINISHED LAST MESSAGE?
1213 005064 001375 BNE 115 ;; IF NOT: WAIT
1214 005066 017637 000004 001320 MOV #4(SP), $FATAL ;; 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 PC ;; RETURN
1223 005130 000 SMFLG: .BYTE 0 ;; MESSG. FLAG
1224 005131 000 SLFLG: .BYTE 0 ;; LOG FLAG
1225 005132 000 SFFLG: .BYTE 0 ;; FATAL FLAG
1226 005134 .EVEN
1227 000200 APTSIZE=200
1228 000001 APTENV=001
1229 000100 APTSPool=100
1230 000040 APTCSUP=040
1231 ;
1232

```

.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

1270	005134	011646			
1271	005135	016665	000004	000002	
1272	005136	105777	174074		
1273	005137	100375			
1274	005138	117746	174070	000004	
1275	005139	042766	177600	000004	
1276	005140	024627	000004	000023	
1277	005141	001013			
1278	005142	105777	174042	2S:	
1279	005143	100375			
1280	005144	117746	174036		
1281	005145	042716	177600		
1282	005146	024627	000021		
1283	005147	001366			
1284	005148	000750			
1285	005149	024627	000004	000140	3S:
1286	005150	005407			
1287	005151	024627	000004	000175	
1288	005152	003003			
1289	005153	042766	000040	000004	4S:
1290	005154	000002			

```

SR0CHR: MOV (SP), -(SP)
MOV 4(SP), 2(SP)
1S: TSTB 2STKS
BPL 1S
MOV 2STKB, 4(SP)
BIC #177, 4(SP)
CMP 4(SP), #23
BNE 3S
2S: TSTB 2STKS
BPL 2S
MOV 2STKB, -(SP)
BIC #177, (SP)
CMP (SP)+, #21
BNE 1S
BR 1S
3S: CMP 4(SP), #140
BLT 4S
CMP 4(SP), #175
BGT 4S
BIC #40, 4(SP)
4S: 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:

! * R0LIN
! * 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

1271	005254	010346			
1272	005255	005046			
1273	005256	012703	005510		
1274	005257	022713	005517		
1275	005258	101486			
1276	005259	104402			
1277	005260	112613			
1278	005261	122713	000177		
1279	005262	001022			
1280	005263	006716			
1281	005264	001007			
1282	005310	112737	000134	005506	

```

SR0LIN: MOV R3, -(SP)
CLR -(SP)
1S: MOV #TTYIN, R3
2S: CMP #TTYIN+7, R3
BLOS 4S
R0CHR
MOV (SP)+, (R3)
10S: CMPB #177, (R3)
BNE 5S
TST (SP)
BNE 6S
MOVB #' \, 9S

```

```

! * 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

```

```

1289 005316 104401 005506 TYPE R3,95
1290 005316 012716 177777 MOV R3,1,(SP)
1291 005316 005303 65: DEC R3
1292 005316 001227 005510 CMP R3,#STTYIN
1293 005316 102434 BLO 45
1294 005316 111337 005506 MOVB (R3),95
1295 005316 104401 005506 TYPE 25
1296 005316 000746 BR 25
1297 005316 005716 55: TST (SP)
1298 005316 001406 BEQ 75
1299 005316 112737 000134 005506 MOVB #'',95
1300 005316 104401 005506 TYPE (95)
1301 005316 005016 CLR (95)
1302 005316 122713 000025 75: CMPB #25,(R3)
1303 005316 001003 BNE 85
1304 005316 104401 005517 TYPE SCNTLU
1305 005316 000726 BR 15
1306 005316 122713 000022 85: CMPB #22,(R3)
1307 005316 001011 BNE 35
1308 005316 105013 CLRB (R3)
1309 005316 104401 001313 TYPE SCRLF
1310 005316 104401 005510 TYPE STTYIN
1311 005316 000717 BR 25
1312 005316 104401 001312 45: TYPE SQUES
1313 005316 000712 BR 15
1314 005316 111337 005506 35: MOVB (R3),95
1315 005316 104401 005506 TYPE 95
1316 005316 122723 000015 CMPB #15,(R3)+
1317 005316 001306 BNE 25
1318 005316 105013 CLRB -1(R3)
1319 005316 104401 001314 TYPE BLF
1320 005316 005726 TST (SP)+
1321 005316 012503 MOV (SP)+,R3
1322 005316 011646 MOV (SP),-(SP)
1323 005316 016666 000004 000002 MOV 4(SP),2(SP)
1324 005316 012766 005510 000004 MOV #STTYIN,4(SP)
1325 005504 000002 RTI
1326 005506 000 .95: .BYTE 0
1327 005507 000 .BYTE 0
1328 005510 000007 .BLKB 7
1329 005517 136 006525 000012 SCNTLU: .ASCIZ /1U/<15><12>
1330 005524 043536 005015 000 SCNTLG: .ASCIZ /1G/<15><12>
1331 005531 015 051412 051127 SPSMR: .ASCIZ <15><12>/SMR = /
1332 005536 036440 000040 SPNEW: .ASCIZ / NEW = /
1333 005542 020040 020127
1334 005550 020075 000
1335 .EVEN
1336 .SBTTL READ AN OCTAL NUMBER FROM THE TTY
1337
1338 *****
1339 *THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
1340 *CHANGE IT TO BINARY.
1341 *THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
1342 *OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
1343 *FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
1344 *THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.

```


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

;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.

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

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 000000

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=SROCHR TRAP+2(104402) TTY TYPEIN CHARACTER ROUTINE
SROLIN ;:CALL=SROLIN TRAP+3(104403) TTY TYPEIN STRING ROUTINE
SROCT ;:CALL=SROCT TRAP+4(104404) READ AN OCTAL NUMBER FROM TTY
SCOP1 ;:CALL=SCOP1 TRAP+5(104405) CALL TO LOOP ON CURRENT DATA HANDLER
SAVOS ;:CALL=SAVOS TRAP+6(104406) CALL TO REGISTER SAVE ROUTINE
RESIS ;:CALL=RESIS 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
RORCLK ;:CALL=RORCLK 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=SINPUT TRAP+15(104415) CALL TO OCTAL # INPUT ROUTINE
CNVRT ;:CALL=CNVRT TRAP+16(104416) CALL TO
CNVRT ;:CALL=CNVRT TRAP+17(104417) CALL TO


```

1625 006744 005737 006754  TYPDAT: TST      DATAPP      : DATA TABLE?
1626 006750 001402          BEQ      RESREG      : BR IF NO.
1627 006752 104416          CONVRT          : SHOW
1628 006754 000000          DATAPP: 0          : DATA TABLE
1629 006756 104407          RESREG: RES05     : RESTORE PROC REGISTERS
1630 006760 122737 000001 001336 HALTS:  CNPB      @APTENV,SENV : IS APT RUNNING?
1631 006762 001007          BNE          35      : SKIP APT CALL IF NOT.
1632 006764 113737 001214 007002 ROVB      $ITEMB,6S : COPY ERROR #.
1633 006766 004737 004704          JSR      PC,SATY4  : CALL APT SERVICES.
1634 007002 000000          .WORD     0          : ERROR # GOES HERE.
1635 007004 000777          BR        95          : LOCK HERE.
1636 007006 022737 004070 000042 35:      @SENDAD,@#42 : IF ACT-II AUTOMATIC MODE, HALT!!
1637 007014 001403          IS:      15          : HALT ON ERROR?
1638 007016 005777 172216          TST      @SMR      : BR IF NO HALT ON ERROR
1639 007018 100005          BPL      EXITER    : SAVE R0
1640 007020 010046          15:      PUSHRO    : SHOW ERROR PC IN DATA LIGHTS
1641 007022 016600 000002          MOV      2(SP),R0 : HALT
1642 007024 000000          HALT          : GET R0
1643 007026 012500          POPRO       : UPDATE ERROR COUNT
1644 007028 002237 001212 EXITER: INC      SERTTL  : GOTO TOP OF TEST?
1645 007030 032777 000400 172170 BIT      @SM08,@SMR : BR IF YES
1646 007032 001004          IS          15      : GOTO NEXT TEST?
1647 007034 032777 002000 172160 BIT      @SM10,@SMR : BR IF NO
1648 007036 001400          BEQ      25          : SET FOR NEXT TEST
1649          .          : NEXT, @LPADR
1650 007062          15:      MOV      @STACK,SP : RESET SP
1651          .          : @LPADR
1652          25:      RTI          : GOTO SPECIFIED TEST
1653          SRTAB0: 1          : @LPADR
1654 007066          .BYTE   6,2
1655 007070 001460          ERTAB1: SAVPC
1656 007072 000001          XTSTN: 1
1657 007074          .BYTE   3,2
1658 007076 001202          $TSTNM
1659          ;ENTER HERE ON POWER FAILURE
1660          ;
1661          .SBTTL  POWER DOWN AND UP ROUTINES
1662          ;
1663          ;*****
1664          ; POWER DOWN ROUTINE
1665          ;*****
1666 007100 012737 007270 000024 $PMRDN: MOV      @ILLUP,@PMRVEC ; SET FOR FAST UP
1667 007106 012737 000340 000026          MOV      @#40,@PMRVEC+2 ; @P10:7
1668 007114 010046          MOV      R0,-(SP) ; PUSH R0 ON STACK
1669 007116 010146          MOV      R1,-(SP) ; PUSH R1 ON STACK
1670 007120 010246          MOV      R2,-(SP) ; PUSH R2 ON STACK
1671 007122 010346          MOV      R3,-(SP) ; PUSH R3 ON STACK
1672 007124 010446          MOV      R4,-(SP) ; PUSH R4 ON STACK
1673 007126 010546          MOV      R5,-(SP) ; PUSH R5 ON STACK
1674 007130 017746 172104          MOV      @SMR,-(SP) ; PUSH @SMR ON STACK
1675 007134 010637 007274          MOV      SP,@SAVR6 ; SAVE SP
1676 007140 012737 007152 001          MOV      @PMRUP,@PMRVEC ; SET UP VECTOR
1677 007146 000000          HALT
1678 007150 000776          BR        .-2          ;; HANG UP
1679          ;*****
1680          ;*****

```

```

1681 :POWER UP ROUTINE
1682 007152 012737 007270 000024 $PMRUP: MOV $SILLUP, @PMRVEC ;SET FOR FAST DOWN
1683 007160 013706 007274 :SSAVR6, SP ;GET SP
1684 007164 005037 007274 CLR $SAVR6 ;WAIT LOOP FOR THE TTY
1685 007170 005237 007274 15: INC $SAVR6 ;WAIT FOR THE INC
1686 007174 001375 BNE IS ;OF WORD
1687 007176 104401 TYPE ,MPFAIL
1688 007208 104417 CNVRT ,PFTAB
1689 007206 105037 001203 CLRB $ERFLG ;CLEAR ERROR FLAG.
1690 007212 005037 001216 CLR $ERRPC ;CLEAR LAST ERROR PC
1691 007216 013701 002066 MOV $KCSR, R1 ;RESTORE DEVICE ADDRESS.
1692 007228 005011 CLR (R1) ;CLEAR THE CSR.
1693 007234 104410 MSTCLR
1694 007236 012677 172006 MOV (SP)+, @SWR ;POP STACK INTO @SWR
1695 007238 012605 MOV (SP)+, R5 ;POP STACK INTO R5
1696 007234 012604 MOV (SP)+, R4 ;POP STACK INTO R4
1697 007236 012603 MOV (SP)+, R3 ;POP STACK INTO R3
1698 007240 012602 MOV (SP)+, R2 ;POP STACK INTO R2
1699 007242 012601 MOV (SP)+, R1 ;POP STACK INTO R1
1700 007244 012600 MOV (SP)+, R0 ;POP STACK INTO R0
1701 007246 012737 007100 000024 MOV $PMRDN, @PMRVEC ;SET UP THE POWER DOWN VECTOR
1702 007254 012737 000340 000026 MOV @340, @PMRVEC+2 ;PRIO:7
1703 007262 104401 TYPE ;REPORT THE POWER FAILURE
1704 007264 007534 $PMRNG: .WORD MPFAIL ;POWER FAIL MESSAGE POINTER
1705 007266 000002 RTI
1706 007270 000000 $SILLUP: HALT ;THE POWER UP SEQUENCE WAS STARTED
1707 007272 000776 BR -.2 ;BEFORE THE POWER DOWN WAS COMPLETE
1708 007274 000000 $SAVR6: 0 ;PUT THE SP HERE
1709
1710 007276 000001 PFTAB: 1
1711 007300 003 002 .BYTE 3,2
1712 007302 001202 $1STNM
1713
1714 007304 .DELAY:
1715 007304 012777 000020 172562 MOV @20, @KMP04
1716 007312 104412 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
1717 007314 121111 121111 ;POKE CLOCK DELAY BIT
1718 007316 15:
1719 007316 104412 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
1720 007320 121224 121224 ;PORT4+IBUS#11
1721 007322 032777 000020 172544 BIT @BIT4, @KMP04 ;IS CLOCK BIT SET?
1722 007330 001772 BEQ IS ;BR IF NO
1723 007332 000002 RTI
1724
1725 007334 .MSTCLR:
1726 007334 152777 000100 172526 BISB @BIT6, @KMSCRH ;SET MASTER CLEAR
1727 007342 142777 000300 172520 BICB @BIT6:BIT7, @KMSCRH ;CLEAR MASTER CLEAR AND RUN
1728 007350 000002 RTI ;RETURN
1729
1730 007352 .ROMCLK:
1731 007352 152777 000002 172510 BISB @BIT1, @KMSCRH ;SET ROMI
1732 007360 013677 172512 MOV @2(SP)+, @KMP06 ;LOAD INSTRUCTION IN SEL6
1733 007364 062746 000002 ADD @2, -(SP) ;ADJUST STACK
1734 007370 032777 000100 171642 BIT @SW06, @SWR ;HALT IF SW06 =1
1735 007376 001401 BEQ IS ;BR IF SW06 =0
1736 007400 000000 HALT ;HALT BEFORE CLOCKING INSTRUCTION
    
```

POWER DOWN AND UP ROUTINES

```

1737 007402 152777 000003 172460 1S: BISR #BIT1:BIT0,2KMC5RH ;CLOCK INSTRUCTION
1738 007410 152777 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: BISR #BIT4,2KMC5RH ;SET STEP LU
1745 007436 027777 172424 172422 CAP 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 1S ;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 #2,2KNP04 ;IS PGM CLOCK BIT CLEAR?
1759 007506 001772 BEQ 1S ;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 #2,2KNP04 ;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 1S ;BR IF NOT DONE
1767 007532 000002 RTI ;RETURN
1768
1769 007534 050200 051127 043040 MFAIL: .ASCIZ <200>/PMR FAILED. RESTART AT TEST /
(2) 007572 042600 042116 050040 MPASS: .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 MVEC: .ASCIZ /VEC: /
(2) 007733 120 051501 042523 MPASSX: .ASCIZ /PASSES: /
(2) 007744 051105 047522 051522 MERRX: .ASCIZ /ERRORS: /
(2) 007755 124 051505 020124 MTSTN: .ASCIZ /TEST NO: /
(2) 007767 052 000 MASTEX: .ASCIZ /#/
(2) 007771 200 042523 020124 MHEM: .ASCIZ <200>/SET SWITCH REG TO KMC11'S DESIRED ACTIVE./
(2) 010044 041520 020072 000 MERRPC: .ASCIZ /PC: /
(2) 010051 200 020040 020040 XHEAD: .ASCII <200>/
(2) 010110 020200 020040 020040 .ASCII <200>/
(2) 010147 200 020040 041520 .ASCII <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 000 CSR: .ASCIZ <200>/CSR ADDRESS?/
(2) 010353 200 042526 000 VEC: .ASCIZ <200>/VECTOR ADDRESS?/
(2) 010374 041200 020122 000 PRIO: .ASCIZ <200>/BR PRIORITY LEVEL? (4,5,6,7)?/
(2) 010433 200 044127 000 .ASCIZ <200>/WHICH LINE UNIT? IF NONE TYPE "N", IF M8201 TYPE "1", IF M8202 TYP
(2) 010545 200 053520 000 .ASCIZ <200>/SWITCH PAC#1 (DDCMP LINE #)?/

```

POWER DOWN AND UP ROUTINES

```

(2) 010603      200 057523 052111  BR:      .ASCIZ  <200>/SWITCH PGM2 (BM873 BOOT ADD)?/
(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  COVER:   .ASCIZ  <200><200>/KMC11 AT NONSTANDARD ADDRESS PC: /
(2) 011001      200 054105 042520  CERR:    .ASCIZ  <200>/EXPECTED FOUND/
(2) 011022      024040 046513 024503  KMC:     .ASCIZ  / (KMC) /
(2) 011032      044600 046114 043505  MINT:    .ASCIZ  <200>/ILLEGAL INTERRUPT ERROR/
(2) 011064      011064  .EVEN
(2) 011064      000005  XSTAT9:  5
1770 011066      006      003      .BYTE    6,3
1771 011070      001276  STMP0
1772 011072      006      003      .BYTE    6,3
1773 011074      001300  STMP1
1774 011076      006      003      .BYTE    6,3
1775 011100      001302  STMP2
1776 011102      006      003      .BYTE    6,3
1777 011104      001304  STMP3
1778 011106      006      002      .BYTE    6,2
1779 011110      001306  STMP4
1780          .EVEN
1781          ;BUFFERS FOR INPUT-OUTPUT
1782
1783
1784 011112      000000  TEMP:    0
1785          .=. +40
1786 011154      000000  MDATA:   0
1787          .=. +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, @ENV        ; IS IT RUNNING UNDER RPT?
1797 011234      001071  BNE     CKSMRS           ; EXIT IF YES.
1798 011236      022777 000007 170002  CMP     @7, @STKB       ; WAS CTRL G TYPED? (7 BIT ASCII)
1799 011244      001404  BEQ     @1              ; BR IF YES
1800 011246      022777 000207 167772  CMP     @207, @STKB     ; 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, @SFLG     ; 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  SOFTSM ; OF SOFT SWITCH REGISTER
1811 011310      104401 005542  CKSMR3: TYPE   @SNEW   ; 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     @5            ; BR IF YES
1815 011326      022703 000012  CMP     @12, R3       ; WAS IT A LF?
1816 011332      001416  BEQ     @5            ; BR IF YES

```


1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907

011464 005737 001470
011470 001004
011472 104401 010703
011476 000000
011476 000776
011480 000241
011504 005137 001500
011510 005537 001500
011514 002737 000004 001504
011520 002737 000010 001502
011520 002737 002300 001502
011520 001006
011540 012737 002100 001502
011546 012737 002302 001504
011554 033737 001500 001470
011562 001747
011564 013700 001502
011570 013702 001504
011574 012037 002066
011580 011037 002056
011604 042737 177000 002056
011612 012037 002050
011616 012037 002052
011622 012037 002054
011626 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 KMACTV ; ARE ANY KMC11'S TO BE TESTED?
BNE 15 ; 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
ROR RUN ; CATCH CARRY FROM RUN
ADD #4,MILK ; UPDATE POINTER
ADD #10,CREAM ; UPDATE ADDRESS POINTER.
CMP #K1.MAP+200,CREAM
BNE 25 ; KEEP GOING; NOT ALL TESTED FOR.
MOV #K01.MAP,CREAM ; RESET ADDRESS POINTER.
MOV #CNT.MAP,MILK ; RESET PASS COUNT POINTER
25: BIT RUN,KMACTV ; IS THIS ONE ACTIVE?
BNE 15 ; BR IF NO
MOV CREAM,R0 ; GET ADDRESS POINTER
MOV MILK,R2 ; GET PASS COUNT POINTER
MOV (R0)+,KMC1R ; LOAD SYSTEM CTRL. REG
MOV (R0),KMRVEC ; LOAD VECTOR
BIC #177000,KMRVEC ; CLEAR UNWANTED BITS
MOV (R0)+,STAT1 ; LOAD STAT1
MOV (R0)+,STAT2 ; LOAD STAT2
MOV (R0)+,STAT3 ; LOAD STAT3
MOV (R2)+,ERRSS ; LOAD PASS COUNT
MOV (R2)+,ERRCTL ; LOAD ERROR COUNT
MOV #2,R0 ; SAVE CORE THIS WAY!
MOV KMC1R,KMC1RH
INC KMC1RH
MOV KMC1RH,KMC1L
INC KMC1L
MOV KMC1L,KMP04
ADD R0,KMP04
MOV KMP04,KMP06
ADD R0,KMP06
MOV KMRVEC,KMRLVL ; PTY LVL
ADD R0,KMRLVL ; TX VEC
MOV KMRLVL,KMTVEC ; TX LVL
ADD R0,KMTVEC
MOV KMTVEC,KMTLVL
ADD R0,KMTLVL
MOV #177000,KMP04 ; SET UP INTERRUPT VECTORS.
MOV #PRE,KMP04 ; SET PRIORITY 5.
MOV #OUT,KMTVEC ; SET UP INTERRUPT VECTORS.
MOV #177000,KMP06 ; SET PRIORITY 5.
MOV #177000,KMP06 ; SET MAIN PROGRAM PRIORITY AS 4.

```

1908 ;SOTHAT KMC CAN INTERRUPT...
1909
1910 012006 032737 000002 001446 BIT #SM01,STRTSW ;IS TEST NO. SELECTED
1911 012014 001447 BEQ 75 ;BR IF NO
1912 4S: 012016 005737 000042 TST 2042 ;RUNNING IN AUTO MODE?
1913 012016 005737 000042 BNE 75 ;BR IF YES
1914 012022 001044 TYPE ,SCLF
1915 012024 104401 001313 INPUT
1916 012030 104415 HTSTN
1917 012032 007755 I
1918 012034 000001 I000
1919 012036 001000 $STNM
1920 012038 001202 .BYTE 0
1921 012040 000 .BYTE 1
1922 012042 001
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 $STNM,2(RO) ;DOES $STNM MATCH?
1928 012058 001014 BNE 6S ;BR IF NO
1929 012060 022760 001202 000004 CMP #STNM,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 #SM01,STRTSW
1934 012114 000412 BR 8S
1935 012116 005720 6S: TST (RO)+ ;POP RO
1936 012120 000027 034324 CMP RO,#TLAST+10 ;AT END YET?
1937 012124 001351 BNE 5S ;BR IF NO
1938 012126 104401 001312 TYPE ,SQUES ;YES ILLEGAL TEST NO.
1939 012132 000731 BR 4S ;TRY AGAIN
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 @SLPADR ;GO START TESTING.
1944
1945
1946 ;ROUTINE USED TO "AUTO SIZE" THE KMC11
1947 ;CSR AND VECTOR.
1948 ;NOTE: THE CSR MAY BE ANY WHERE IN THE FLOATING
1949 ; AND THE VECTOR MAY BE ANY WHERE IN THE
1950 ; FLOATING VECTOR RANGE (300:770)
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963

```

```

AUTO.SIZE:
CSRMAP: RESET ;INSURE A BUS INIT.
MOV #KCH.MAP,R2 ;LOAD MAP POINTER.
1S: CLR (R2)+ ;ZERO ENTIRE MAP
CMP #KCH.END,R2 ;ALL DONE?
BNE 1S ;BR IF NO
CLR #KCHUM ;SET OCTAL NUMBER OF KMC11'S TO 0
MOV #KCH.MAP,R2 ;R2 POINTS TO KMC MAP
CLR #KMACTV ;CLEAR ACTIVE
BIT #SM00,STRTSW ;QUESTIONS?

```

```

1964 012212 001002 ONE 16 ;BB IF YES
1965 012214 000137 JMP 75 ;IF NO SKIP QUESTIONS
1966 012216 012737 012574 001306 MOV #1,STMP4 ;START WITH 1
1967 012218 104415 INPUT NUM
1968 012220 010275 NUM
1969 012222 000001 I
1970 012224 000020 I6.
1971 012226 001302 STMP2
1972 012228 000 .BYTE 0
1973 012230 001 .BYTE 1
1974 012232 013737 001302 001472 MOV STMP2,KNUM ;KNUM = HOW MANY
1975 012234 104401 001313 12S: TYPE ,SCRLF ;TYPE WHICH KMC IS BEING DONE
1976 012236 104415 CONVERT ;STMP4 IS WHICH KMC
1977 012238 012225 WHICH
1978 012240 005237 001306 INC STMP4
1979 012242 104415 INPUT
1980 012244 010353 CSR
1981 012246 160000
1982 012248 164000
1983 012250 001304 STMP3
1984 012252 000 .BYTE 0
1985 012254 001 .BYTE 1
1986 012256 013722 001304 MOV STMP3,(R2)+ ;STORE CSR IN MAP
1987 012258 104415 INPUT
1988 012260 010353 VEC
1989 012262 000000
1990 012264 000776
1991 012266 001304 STMP3
1992 012268 000 .BYTE 0
1993 012270 001 .BYTE 1
1994 012272 013712 001304 MOV STMP3,(R2) ;STORE VECTOR IN MAP
1995 012274 104401 TYPE
1996 012276 010374 10S: Prio
1997 012278 004737 013520 JSR PC,INTTY ;ASK WHAT BR LEVEL
1998 012280 022703 000024 CMP #24,R3 ;GET RESPONSE
1999 012282 101014 BHI $05 ;BR IF LESS THAN 4
2000 012284 022703 000027 CMP #27,R3 ;BR IF GREATER THAN 7
2001 012286 103411 BLO $05 ;R4 = NUMBER OF SHIFTS
2002 012288 012704 000011 MOV #1,R4 ;SHIFT R3 LEFT
2003 012290 005303 RSL R3 ;DEC SHIFT COUNT
2004 012292 005304 DEC R4 ;BR IF NOT DONE
2005 012294 001375 BNE .4 ;BIC UNWANTED BITS
2006 012296 042703 170777 BIC #170777,R3 ;PUT BR LEVEL IN STATUS MAP
2007 012298 050312 BIS R3,(R2) ;CONTINUE
2008 012300 000403 BR $5
2009 012302 104401 50S: TYPE
2010 012304 001312 SQUES ;RESPONSE IS OUT OF LIMITS
2011 012306 000752 BR 10S ;TRY AGAIN
2012 012400 8S:
2013 012400 9S:
2014 012400 16S: TYPE
2015 012402 104401 MODU ;ASK WHICH LINE UNIT
2016 012404 004737 013520 JSR PC,INTTY ;GET REPLY
2017 012410 022703 000021 CMP #21,R3 ;"1"
2018 012414 001417 BEQ $05
2019 012416 022703 000022 CMP #22,R3 ;"2"
    
```

POWER DOWN AND UP ROUTINES

2020	012422	001412			BEQ	315		
2021	012424	022703	000116		CFB	#116,R3	;	"N"
2022	012426	001403			BEQ	325		
2023	012428	104401			TYPE			
2024	012429	001312			SOLES			; IF NOT A 1,2 OR N TYPE ""
2025	012436	000760			BR	165		; TRY AGAIN
2026	012438	052722	010000	325:	BIS	#BIT12,(R2)+		; SET BIT 12 IN STAT2 IF NO LU
2027	012441	022722			CFB	(R2)+,(R2)+		; POP OVER STAT2 AND STAT3
2028	012446	000445			BR	335		
2029	012450	052712	020000	315:	BIS	#BIT13,(R2)		; SET BIT 13 IN STAT2 IF M8202
2030	012454	104401		305:	TYPE			
2031	012456	010543			CONN			; ASK IF LOOP-BACK IS ON
2032	012460	004737	013520		JSR	PC,INTTY		; GET REPLY
2033	012464	022703	000131		CFB	#131,R3		; Y
2034	012470	001406			BEQ	175		
2035	012472	022703	000116		CFB	#116,R3		; N
2036	012476	001406			BEQ	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	012535	104415			INPUT			
2053	012537	010603			BM			
2054	012538	000000			0			
2055	012539	000377			377			
2056	012540	001304			STMP3			
2057	012542	000			.BYTE	0		
2058	012543	001			.BYTE	1		
2059	012544	113722	001304		MOVB	STMP3,(R2)+		; STORE SWITCH PAC IN MAP
2060	012546	052722			TST	(R2)+		; POP OVER STAT3
2061	012547	005337	001302	335:	DEC	STMP2		; DEC KMC COUNT
2062	012548	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	012600	012737	013220	000004	MOV	#65,204		; SET FOR NON-EXISTANT DEVICE TIME OUT
2066	012606	005011		25:	CLR	(R1)		; CLEAR SEL0
2067	012610	005711			TST	(R1)		; IF KMC11 KMC5R 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 KMC1R 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!

POWER DOWN AND UP ROUTINES

```

2075 012660 002761 125252 000004      CMP      #125252,4(R1)      ;WAS IT WRITTEN?
2076 012666 001113                    BNE      #38                ;IF NO IT IS NOT CRAM
2077 012672 001113                    ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A KMC11 CSR ADDRESS.
2078 012678 001113                    ;
2079 012684 001113                    ;
2080 012690 001113                    ;
2081 012696 001113                    ;
2082 012702 001113                    ;
2083 012708 001113                    ;
2084 012714 001113                    ;
2085 012720 001113                    ;
2086 012726 001113                    ;
2087 012732 001113                    ;
2088 012738 001113                    ;
2089 012744 001113                    ;
2090 012750 001113                    ;
2091 012756 001113                    ;
2092 012762 001113                    ;
2093 012768 001113                    ;
2094 012774 001113                    ;
2095 012780 001113                    ;
2096 012786 001113                    ;
2097 012792 001113                    ;
2098 012798 001113                    ;
2099 012804 001113                    ;
2100 012810 001113                    ;
2101 012816 001113                    ;
2102 012822 001113                    ;
2103 012828 001113                    ;
2104 012834 001113                    ;
2105 012840 001113                    ;
2106 012846 001113                    ;
2107 012852 001113                    ;
2108 012858 001113                    ;
2109 012864 001113                    ;
2110 012870 001113                    ;
2111 012876 001113                    ;
2112 012882 001113                    ;
2113 012888 001113                    ;
2114 012894 001113                    ;
2115 012900 001113                    ;
2116 012906 001113                    ;
2117 012912 001113                    ;
2118 012918 001113                    ;
2119 012924 001113                    ;
2120 012930 001113                    ;
2121 012936 001113                    ;
2122 012942 001113                    ;
2123 012948 001113                    ;
2124 012954 001113                    ;
2125 012960 001113                    ;
2126 012966 001113                    ;
2127 012972 001113                    ;
2128 012978 001113                    ;
2129 012984 001113                    ;
2130 012990 001113                    ;
2131 012996 001113                    ;

```

```

219:      MOV      R1,(R2)+      ;STORE CSR IN CORE TABLE.
220:      MOV      #BIT9,(R1)   ;CLEAR LINE UNIT LOOP
155:      CLR      4(R1)        ;CLEAR PORT4
      MOV      #122113,6(R1) ;LOAD INSTRUCTION (CLR DTR)
      BIS      #BIT8,(R1)    ;CLOCK INSTRUCTION
      MOV      #021254,6(R1) ;LOAD INSTRUCTION
      BIS      #BIT8,(R1)    ;CLOCK INSTRUCTION
      CMB      #377,4(R1)    ;IS IT ALL ONES?
      BNE      #-10          ;OR IF NO
      BIS      #BIT12,(R2)   ;IF YES, NO LINE UNIT, SET STATUS BIT
      BR      208
      BIT      #BIT1,4(R1)   ;IS SWITCH A ONE?
      BEQ      #-10          ;OR IF NO
      BIS      #BIT13:BIT14,(R2) ;NEED2 ASSUME CONNECTOR
      BR      208           ;CONNECTOR ON)
      BIT      #BIT3,4(R1)   ;IS HWY SET
      BNE      #-10          ;OR IF NO
      MOV      #BIT6,4(R1)   ;LOAD PORT4
      MOV      #122113,6(R1) ;LOAD INSTRUCTION
      BIS      #BIT8,(R1)    ;CLOCK INSTRUCTION(SET DTR)
      MOV      #021254,6(R1) ;LOAD INSTRUCTION
      BIS      #BIT8,(R1)    ;CLOCK INSTRUCTION(READ MODEM REG)
      BIT      #BIT3,4(R1)   ;IS HWY SET NOW?
      BEQ      208          ;OR IF NO CONNECTOR
      BIS      #BIT14,(R2)   ;SET STATUS BIT FOR CONNECTOR
205:      TST      (R2)+        ;POP POINTER
      MOV      #021324,6(R1) ;PUT INSTRUCTION IN PORT6
      MOV      #BIT9:BIT8,(R1) ;PORT4->LJ IS
      BISH      4(R1),(R2)+ ;STORE DDCP LINE # IN TABLE
      MOV      #021344,6(R1) ;PORT6->INSTRUCTION
      MOV      #BIT8:BIT9,(R1) ;CLOCK INSTR.
      BISH      4(R1),(R2)+ ;STORE #0073 ADD IN TABLE
      TST      (R2)+        ;POP OVER STAT3
      CLR      (R1)         ;CLEAR NOM1
      INC      KINUM        ;UPDATE DEVICE COUNTER
      CMP      #20,KINUM    ;ARE MAX. NO. OF DEV FOUND?
      BEQ      138         ;YES DON'T LOOK FOR ANY MORE.
35:      CLR      (R1)       ;CLEAR BIT 10
      CLR      6(R1)       ;CLEAR SEL 6
145:      ADD      #10,R1    ;UPDATE CSR POINTER ADDRESS
      CMP      #164000,R1  ;BR IF MORE ADDRESS TO CHECK.
      BNE      28
135:      CLR      KMACTV   ;WERE ANY KMC11'S FOUND AT ALL?
      TST      KINUM       ;ERROR AUTO SIZER FOUND NO KMC11'S IN THIS SYS.
      BEQ      58
      MOV      KINUM,R1
      MOV      R1,SAVNUM   ;SAVE NUMBER OF DEVICES
45:      CLC
      ROL      KMACTV     ;GENERATE ACTIVE REGISTER OF DEVICES.
      INC      KMACTV     ;SET THE BIT
      DEC      R1
      BNE      45        ;BR IF MORE TO GENERATE

```

POWER DOWN AND UP ROUTINES

2132	013166	012737	000006	000004		MOV	#6, R4	RESTORE TRAP VECTOR
2133	013174	013737	001470	001474		MOV	KRACTV, SAVACT	SAVE ACTIVE REGISTER
2134	013202	000137	013234			JMP	VECMAP	GO FIND THE VECTOR NOW.
2135	013206	104401	007617		55:	TYPE	HERR2	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. SM.
2139	013220	012716	013114		65:	MOV	#145, (SP)	ENTERED BY NON-EXISTANT TIME-OUT.
2140	013224	000002				RTI		RETURN TO MAINSTREAM
2141	013226	000001						
2142	013230	002	0			.BYTE	2,2	
2143	013232	001306				STMP4		
2144	013234	032737	000001	001	ECHMAP:	BIT	#SM00, STRTSM	
2145	013242	001114				BNE	55	
2146	013244	012737	000340	000		MOV	#340, J822	SET IOT TRAP PRIO TO 7
2147	013250	012737	013426	00002C		MOV	#45, J820	SET IOT TRAP VECTOR
2148	013250	012702	002100			MOV	#01, MAP, R2	SET SOFTWARE POINTER
2149	013254	012700	000300			MOV	#300, R0	FLOATING VECTORS START HERE.
2150	013270	012701	000302			MOV	#302, R1	PC OF IOT INSTR.
2151	013274	010120			15:	MOV	R1, (R0)+	START FILLING VECTOR AREA
2152	013276	012721	000004			MOV	#4, (R1)+	WITH .+2; IOT
2153	013280	002021				CMP	(R0)+, (R1)+	ADD 2 TO R0 +R1
2154	013304	000127	001000			CMP	R1, #1000	
2155	013310	101771				BLOS	15	BR IF MORE TO FILL
2156	013312	013737	001470	001276		MOV	KRACTV, STMP0	STORE TEMPORALLY
2157	013320	005007	001276		25:	ROR	STMP0	BRING OUT A BIT
2158	013324	102003				BCC	55	BR IF ALL DONE
2159	013326	012704	000012			MOV	#12, R4	R4 IS INDEX REGISTER
2160	013328	016437	013504	177776		MOV	BRVL(R4), PS	SET PS TO 7
2161	013330	011201				MOV	(R2), R1	
2162	013340	012761	000200	000004		MOV	#200, 4(R1)	
2163	013360	012711	001000			MOV	#BIT9, (R1)	SET ROMI
2164	013364	012761	121111	000006		MOV	#121111, 6(R1)	PUT INSTRUCTION IN PORT6
2165	013364	012711	001400			MOV	#BIT9, BIT8, (R1)	FORCE AN INTERRUPT
2166	013366	105200			75:	INCB	R0	STALL
2167	013370	001376				BNE	.-2	FOR TIME TO INTERRUPT
2168	013372	162704	000002			SUB	#2, R4	GET NEXT LOWEST PS LEVEL
2169	013376	001404				BGE	65	BR IF R4 = 0
2170	013400	016437	013504	177776		MOV	BRVL(R4), PS	MOVE NEXT LOWER LEVEL IN PS
2171	013406	000767				BR	75	BR TO DELAY
2172	013410	002762	005300	000002	65:	BIS	#5300, 2(R2)	NO INTERRUPT ASSUME 300 AT LEVEL 5 AND FIX KMC11 LATER
2173	013416	005011			35:	CLR	(R1)	CLEAR ROMI
2174	013420	002702	000010			ROR	#10, R2	POP SOFTWARE POINTER
2175	013424	000736				BR	25	KEEP GOING
2176	013426	051642	000002		45:	BIS	(SP), 2(R2)	GET VECTOR ADDRESS
2177	013428	042762	000007	000002		BIC	#7, 2(R2)	CLEAR JUNK
2178	013430	016405	013506			MOV	BRVL+2(R4), R5	GET BR LEVEL OF KMC11
2179	013434	005305				REL	R5	SHIFT LEVEL 4 PLACES
2180	013436	005305				REL	R5	TO THE LEFT FOR THE
2181	013438	005305				REL	R5	STATUS TABLE
2182	013438	005305				REL	R5	
2183	013454	042705	170777			BIC	#170777, R5	CLEAR UNWANTED BITS
2184	013460	050562	000002			BIS	R5, 2(R2)	PUT BR LEVEL IN STATUS TABLE
2185	013464	022626				CMP	(SP)+, (SP)+	POP IOT JUNK OFF STACK

DZKCA MACY11 27(1006) 13-MAY-77 14:07 PAGE 46
DZKCA.P11 13-MAY-77 13:58 POWER DOWN AND UP ROUTINES

```

2188 013766 012716 013416      MOV      #35,(SP)      ;SET FOR RETURN
2189 013772 000002      RTI
2190 013774 012737 004134 000020 55:  MOV      @SSCOPE,@#20  ; RESTORE SCOPE VECTOR
2191 013802 000207      RTS      PC           ;ALL DONE WITH "AUTO SIZING"

2192 013804 000000      BRLVL:  PRO        ;LEVEL 0
2193 013806 000000      PRO        ;LEVEL 0
2194 013810 000200      PR4        ;LEVEL 4
2195 013812 000240      PR5        ;LEVEL 5
2196 013814 000300      PR6        ;LEVEL 6
2197 013816 000340      PR7        ;LEVEL 7

2200 013820 105777 165520      INTTY:  TSTB      @STKS      ;WAIT FOR DONE
2201 013822 100375      BPL      -4
2202 013824 017703 165514      MOV      @STKB,R3      ;PUT CHAR IN R3
2203 013826 105777 165512      TSTB      @STPS      ;WAIT UNTIL PRINTER IS READY
2204 013828 100375      BPL      -4
2205 013830 010377 165506      MOV      R3,@STPB      ;ECHO CHAR
2206 013832 012703 000240      BIC      @BIT7!BITS,R3 ;MASK OFF LOWER CASE
2207 013850 000207      RTS      PC           ;RETURN

2210 013850      APT.SIZE:
2211 013852 000005      RESET
2212 013854 010046      MOV      R0,-(SP)      ;: PUSH R0 ON STACK
2213 013856 010146      MOV      R1,-(SP)      ;: PUSH R1 ON STACK
2214 013858 010246      MOV      R2,-(SP)      ;: PUSH R2 ON STACK
2215 013859 010346      MOV      R3,-(SP)      ;: PUSH R3 ON STACK
2216 013864 005037 013766      CLR      VECTR        ;: CLEAR THE LOCAL VARIABLE
2217 013870 005037 013772      CLR      PRITY        ;: CLEAN UP LOCAL VARIABLE
2218 013874 013700 001376      MOV      SCOM1,R0      ;: GET THE DEVICE COUNT
2219 013880 010037 001476      MOV      R0,SAVNUM     ;: SAVE THE NO. OF DEVICES
2220 013884 012701 001346      MOV      @SAVMS1,R1    ;: GET EXTRA INFO. BITS POINTER
2221 013890 013737 001372 013770      MOV      @BASE,BASE    ;: GET BASE CSR ADDRESS
2222 013896 113737 001366 013766      MOVVB   SVECT1,VECTR   ;: GET THE VECTOR
2223 013902 113737 001367 013772      MOVVB   SVECT1+1,PRITY ;: GET THE PRIORITY
2224 013908 013737 001374 001470      MOV      @DEVH,KRACTV  ;: SAVE THE KAC'S SELECTED ACTIVE
2225 013914 013737 001470 001474      MOV      @KRACTV,SAVACT ;: SAVE THE ACTIVE REGISTER
2226 013920 012702 001402      MOV      @@0000,R2     ;: GET ADDRESS OF FIRST DEVICE DESCRIPTOR WORD
2227 013926 012703 002100      MOV      @K01.MAP,R3   ;: GET POINTER TO DEVICE MAP
2228 013932 005023      CLR      (R3)+         ;: CLEAR DEVICE MAP
2229 013938 022703 002300 35:  CMP      @K01.END,R3   ;: IS WHOLE DEV.MAP CLEARED?
2230 013944 003374      BGT      35           ;: NO, THEN GO ON.
2231 013950 012703 002100      MOV      @K01.MAP,R3   ;: RESTORE DEV.MAP POINTER.
2232 013956 013723 013770 15:  MOV      @BASE,(R3)+   ;: LOAD CSR ADDRESS
2233 013962 112163 000001      MOVVB   (R1)+,1(R3)   ;: GET EXTRA INFO. BITS
2234 013968 006213      ASR      (R3)          ;: SET IT IN RIGHT POSITION.
2235 013974 006213      ASR      (R3)          ;: SET IT IN RIGHT POSITION.
2236 013980 053713 013772      BIS     PRITY,(R3)     ;: GET PRIORITY IN STAT1
2237 013986 006313      ASL      (R3)          ;: SET THEM IN RIGHT POSITION
2238 013992 006313      ASL      (R3)          ;:
2239 013998 006313      ASL      (R3)          ;:
2240 014004 006313      ASL      (R3)          ;:
2241 014010 053723 013766      RIS     VECTR,(R3)+   ;: GET THE VECTOR IN STAT1
2242 014016 012223      MOV      (R2)+,(R3)+  ;: GET THE STAT2 FROM DC
2243 014022 005723      TST     (R3)+         ;: SKIP OVER STAT3

```

```

2244 013732 005300          DEC      R0          ; COUNT BY 1
2245 013734 001407          BEQ      25          ; 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       1$         ; SET THE NEXT MAP ENTRY
2249 013754
2250 013754 012603          2$:      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 013763 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          PRIORITY: .WORD    0
2258 013774          SFLT1  OUT1,INP1,4,0
2259 013774          SXZ

;***** TEST 1 *****
; * MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
; * FLOAT A 1 THROUGH REGISTER OUT1 <4>
; * FLOAT A 0 THROUGH REGISTER OUT1 <4>
2265 013774          SXZ
;*****

2268 013774          STSTN
; 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.
; R1 CONTAINS BASE MACY11 ADDRESS
; LOAD-VERIFY-WAIT.
2275 014010 004737 035536          JSR      PC,LDRVMT
2276 014014 014030          MCT1
2277 014016 104022          ERROR   22          ; TIME OUT ERROR...
2278 014020 012706 001200          MOV      #STACK,SP  ; RESET STACK...
2279 014024 000177 165412          JMP      @NEXT      ; GO TO NEXT TEST...
2280 014030          MCT1:
2281 014030          21$:
2282 014030          MOVE     #0,BREG   ; SET TO CLEAR SPAD 16
2283 000000          MICPC=MICPC+1
2284 014030 000400          .WORD   .S.
2285 014032          MOVE     BREG,SPAD <16> ; FOR RETURN ADDRESS PURPOSES...
2286 000001          MICPC=MICPC+1
2287 014032 063236          .WORD   .S.
2288 014034          SFLT0  OUT1,INP1,4,0,1,1$,2$,3$,4$
2289 014034          1$:  MOVE     #200,BREG  ; START WITH BIT 7.
2290 000002          MICPC=MICPC+1
2291 014034 000600          .WORD   .S.
2292 014036          2$:
2293 014036          MOVE     BREG,OUT1 <4> ; SET THE BIT.
2294 000003          MICPC=MICPC+1
2295 014036 061224          .WORD   .S.
2296 014040          MOVE     INP1 <4>,SPAD <0> ; GET THE "FOUND" IN SCRATCH PAD.
2297 000004          MICPC=MICPC+1
2298 014040 123100          .WORD   .S.
2299 014042          MOVE     BREG,SPAD <4>

```

```

2300          000005          MICPC=MICPC+1
2301 014042 063224          .WORD .S.
2302 014044          SIFEQ BREG,SPAD <0> 35          ;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 35
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 ERROR          ;REPORT DATA ERROR.
2330 014064          MOVE # <MICPC+3>,BREG
2331          000016          MICPC=MICPC+1
2332 014064 000420          .WORD .S.
2333 014066          SBR ERROR
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 25          ;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 25          ;YES, DO IT.
2353          000025          MICPC=MICPC+1
2354 014102 100403          .WORD .S.
2355 014104          MOVE SPAD <4>,BREG
    
```

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.
SHFBRT ;NO, CONTINUE...
MICPC=MICPC+1
.WORD .SBR!.SELB!.DBRSH
BB7 45 ;IS IT DONE?...
MICPC=MICPC+1
.WORD .S.
SBR 25 ;NO, CONTINUE...
.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...
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 <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.
.WORD .S.
CALL EROR ;REPORT DATA ERROR.
MOVE # <MICPC+3>,BREG
MICPC=MICPC+1

```

12 014144 000450
13 014146
14 00047
15 014146 104400
16 014150
17 000050
18 014150 060604
19 014150
20 000051
21 01415 100433
22 014155
23 014155
24 000052
25 014155 000454
26 014158
27 000053
28 014158 104427
29 014160
30 014160 060604
31 014160 070055
32 014162 100433
33 014164
34 000056
35 014164 060604
36 014166
37 000057
38 014166 061620
39 014170
40 000060
41 014170 103433
42 014172
43 014172
44 000061
45 014172 000463
46 014174
47 000062
48 014174 104454
49 014176
50 000063
51 014176 100400
52 014200
53 014200

```
.WORD .S.  
SBR EROR  
MICPC=MICPC+1  
MOVE SPAD <4>,BREG ;RESTORE BREG...  
MICPC=MICPC+1  
WORD .S.  
SBR 128 ;LOOP ON ERROR...  
MICPC=MICPC+1  
WORD .S.  
135: 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 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 ;  
SBR 128  
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 218  
MICPC=MICPC+1  
WORD .S.  
SFLT1 OUT1,INP1,5,0  
SXZ  
  
;***** 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,STSNM ; LOAD THE NO. OF THIS TEST
014206 012737 014404 001442 MOV #TST3,NEXT ; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
;LOAD-VERIFY-WAIT.
014214 004737 035536 JSR PC,LDRWT
014220 014234 MCT2: MCT2 ERROR 22 ;TIME OUT ERROR...
014224 012706 001200 MOV #STACK,SP ;RESET STACK...
014230 000177 165206 JMP #NEXT ;GO TO NEXT TEST...
014234 MCT2: 215: MOVE #0,BREG ;SET TO CLEAR SPAD 16
MICPC=MICPC+1
.WORD $.
014234 000000 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES...
014236 000400 MICPC=MICPC+1
.WORD $.
014236 000001 SFLOT OUT1,INP1 5,0,1,15,25,35,45
014240 063236 18: MOVE #200,BREG ;START WITH BIT 7.
MICPC=MICPC+1
.WORD $.
014240 000002 25: MOVE BREG,OUT1 <5> ;SET THE BIT.
MICPC=MICPC+1
.WORD $.
014242 000003 MOVE INP1 <5>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
014244 061225 MICPC=MICPC+1
.WORD $.
014244 000004 MOVE BREG,SPAD <4>
014246 123120 MICPC=MICPC+1
.WORD $.
014246 000005 $IFEQ BREG,SPAD <0> 35 ;CHECK THE DATA...
014250 063224
014250 000006 SUB2C SPAD <0>,BREG,NOP
014250 060360 MICPC=MICPC+1
.WORD $.
014252 000007 BZ 35
MICPC=MICPC+1
.WORD $.
014252 101422 MOVE BREG,OUT1 <CSR4> ;GOOD DATA...
014254 000010 MICPC=MICPC+1
.WORD $.
014256 061224 MOVE INP1 <5>,OUT1 <CSR5>;BAD DATA...
014256 000011 MICPC=MICPC+1
.WORD $.
014256 121125 MOVE #1,MEM ;TYPE OF ERROR...
014260 000012 MICPC=MICPC+1
.WORD $.
014260 002401 MOVE MEM,OUT1 <CSR3> ;
014262 000013 MICPC=MICPC+1
.WORD $.
014262 041223 MOVE #5,MEM ;
014264 000014 MICPC=MICPC+1
.WORD $.
014264 002405 MOVE MEM,OUT1 <CSR7> ;REG. ADDRESS.
014266
    
```

014266	000015	MICPC=MICPC+1	
014270	041227	.WORD .S.	
014270		CALL EROR	;REPORT DATA ERROR.
014270		MOVE # <MICPC+3>,BREG	
014270	000016	MICPC=MICPC+1	
014272	000420	.WORD .S.	
014272		SBR EROR	
014272	000017	MICPC=MICPC+1	
014274	104400	.WORD .S.	
014274		MOVE SPAD <4>,BREG	;RESTORE BREG...
014274	000020	MICPC=MICPC+1	
014276	060604	.WORD .S.	
014276		SBR 25	;LOOP ON ERROR...
014276	000021	MICPC=MICPC+1	
014300	100403	.WORD .S.	
014300		CALL SCP1	;IS LOOP DATA SET.
014300		MOVE # <MICPC+3>,BREG	
014300	000022	MICPC=MICPC+1	
014302	000424	.WORD .S.	
014302		SBR SCP1	
014302	000023	MICPC=MICPC+1	
014304	104427	.WORD .S.	
014304		MOVE SPAD <4>,BREG	
014304	000024	MICPC=MICPC+1	
014306	060604	.WORD .S.	
014306		SBR 25	;YES, DO IT.
014306	000025	MICPC=MICPC+1	
014310	100403	.WORD .S.	
014310		MOVE SPAD <4>,BREG	
014310	000026	MICPC=MICPC+1	
014312	060604	.WORD .S.	
014312		SFBRT	;NO, CONTINUE...
014312	000027	MICPC=MICPC+1	
014314	061620	.WORD SBR!.SELB!.DBRSH	
014314		BB7 45	;IS IT DONE?...
014314	000030	MICPC=MICPC+1	
014316	103432	.WORD .S.	
014316		SBR 25	;NO, CONTINUE...
014316	000031	MICPC=MICPC+1	
014320	100403	.WORD .S.	
014320		45:	
014320		SFLOT OUT1,INP1,5,0,0,115,125,135,145	
014320		115: MOVE # 177,BREG ;START WITH BIT 7.	
014320	000032	MICPC=MICPC+1	
014322	000577	.WORD .S.	
014322		125:	
014322		MOVE BREG,OUT1 <5> ;SET THE BIT.	
014322	000033	MICPC=MICPC+1	
014324	061225	.WORD .S.	
014324		MOVE INP1 <5>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.	
014324	000034	MICPC=MICPC+1	
014326	123120	.WORD .S.	
014326		MOVE BREG,SPAD <4>	
014326	000035	MICPC=MICPC+1	
014328	063224	.WORD .S.	
014330		SIFEQ BREG,SPAD <0> 135 ;CHECK THE DATA...	

014330
014330
014332
014332
014334
014334
014336
014336
014340
014340
014342
014342
014344
014344
014346
014346
014350
014350
014352
014352
014354
014354
014356
014356
014360
014360
014362
014362
014364
014364
014366
014366
014370
014370
014372

000036
060360
000037
101452
000040
061224
000041
121125
000042
002401
000043
041223
000044
002405
000045
041227
000046
000450
000047
104400
000050
060604
000051
100433
000052
000454
000053
104427
000054
060604
000055
100433
000056
060604

135:

```
SUB2C SPAD <0>,BREG,NOP
MICPC=MICPC+1
.WORD $.
BZ 135
MICPC=MICPC+1
.WORD $.
MOVE BREG OUT1 <CSR4> ;GOOD DATA...
MICPC=MICPC+1
.WORD $.
MOVE INP1 <5>,OUT1 <CSRS>;BAD DATA...
MICPC=MICPC+1
.WORD $.
MOVE 8 1, MEM ;TYPE OF ERROR...
MICPC=MICPC+1
.WORD $.
MOVE MEM OUT1 <CSR3> ;
MICPC=MICPC+1
.WORD $.
MOVE 8 5, MEM ;
MICPC=MICPC+1
.WORD $.
MOVE MEM OUT1 <CSR7> ;REG. ADDRESS.
MICPC=MICPC+1
.WORD $.
CALL EROR ;REPORT DATA ERROR.
MOVE 8 <MICPC+3>,BREG
MICPC=MICPC+1
.WORD $.
SBR EROR
MICPC=MICPC+1
.WORD $.
MOVE SPAD <4>,BREG ;RESTORE BREG...
MICPC=MICPC+1
.WORD $.
SBR 125 ;LOOP ON ERROR...
MICPC=MICPC+1
.WORD $.
CALL SCP1 ;IS LOOP DATA SET.
MOVE 8 <MICPC+3>,BREG
MICPC=MICPC+1
.WORD $.
SBR SCP1
MICPC=MICPC+1
.WORD $.
MOVE SPAD <4>,BREG
MICPC=MICPC+1
.WORD $.
SBR 125 ;YES, DO IT.
MICPC=MICPC+1
.WORD $.
MOVE SPAD <4>,BREG
MICPC=MICPC+1
.WORD $.
SHFBRT ;NO, CONTINUE...
```

KMC11 MICRO PROCESSOR IBUS* TESTS

2638
 2639
 2640
 2641
 2642
 2643
 2644
 2645
 2646
 2647
 2648
 2649
 2650
 2651
 2652
 2653
 2654
 2655
 2656
 2657
 2658
 2659
 2660
 2661
 2662
 2663
 2664
 2665
 2666
 2667
 2668
 2669
 2670
 2671
 2672
 2673
 2674
 2675
 2676
 2677
 2678
 2679
 2680
 2681
 2682
 2683
 2684
 2685
 2686
 2687
 2688
 2689
 2690
 2691

```

MICPC=MICPC+1
.WORD .SBR!..SELB!..DBRSH
B87 128
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 21$
MICPC=MICPC+1
.WORD .S.

$FLT1
$XZ
OUT1,INP1,6,0

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

$XZ
;*****

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

MCT3:
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.
$FLOT OUT1,INP1,6,0,1,1$,2$,3$,4$
1$: MOVE # 200,BREG ;START WITH BIT 7.
MICPC=MICPC+1
.WORD .S.

2$: 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

```

```

2699 014450 123140      .WORD      $
2700 014452 000005      MOVE      BREG,SPAD <4>
2701 014453 063224      MICPC=MICPC+1
2702 014454 063224      .WORD      $
2703 014454 063224      $IFEQ    BREG,SPAD <0> 3S      ;CHECK THE DATA...
2704 014454 000006      SUB2C    SPAD <0>,BREG,NOP
2705 014454 060360      MICPC=MICPC+1
2706 014456 000007      .WORD      $
2707 014456 101422      BZ       3S
2708 014456 000010      MICPC=MICPC+1
2709 014456 061224      .WORD      $
2710 014456 061224      MOVE     BREG,OUT1 <CSR4>      ;GOOD DATA...
2711 014456 000011      MICPC=MICPC+1
2712 014456 121145      .WORD      $
2713 014456 000012      MOVE     # 1, MEM
2714 014456 002401      MICPC=MICPC+1      ;TYPE OF ERROR...
2715 014456 000013      .WORD      $
2716 014456 041223      MOVE     MEM,OUT1 <CSR3>      ;
2717 014456 000014      MICPC=MICPC+1
2718 014456 002406      .WORD      $
2719 014456 002406      MOVE     # 6, MEM
2720 014456 000015      MICPC=MICPC+1      ;
2721 014456 041227      .WORD      $
2722 014456 000015      MOVE     MEM,OUT1 <CSR7>      ;REG. ADDRESS.
2723 014456 000015      MICPC=MICPC+1
2724 014456 041227      .WORD      $
2725 014456 000016      CALL     ERROR
2726 014456 000420      MICPC=MICPC+1      ;REPORT DATA ERROR.
2727 014456 000017      MOVE     # <MICPC+3>,BREG
2728 014456 104400      MICPC=MICPC+1
2729 014456 000017      .WORD      $
2730 014456 104400      SBR     ERROR
2731 014456 000017      MICPC=MICPC+1
2732 014456 060604      .WORD      $
2733 014456 060604      MOVE     SPAD <4>,BREG      ;RESTORE BREG...
2734 014456 000020      MICPC=MICPC+1
2735 014456 100403      .WORD      $
2736 014456 100403      SBR     2S      ;LOOP ON ERROR...
2737 014456 000021      MICPC=MICPC+1
2738 014456 100403      .WORD      $
2739 014456 000021      CALL     SCP1      ;IS LOOP DATA SET.
2740 014456 000022      MOVE     # <MICPC+3>,BREG
2741 014456 000424      MICPC=MICPC+1
2742 014456 000424      .WORD      $
2743 014456 104427      SBR     SCP1
2744 014456 000023      MICPC=MICPC+1
2745 014456 104427      .WORD      $
2746 014456 060604      MOVE     SPAD <4>,BREG
2747 014456 060604      MICPC=MICPC+1
2748 014456 000024      .WORD      $
2749 014456 060604      SBR     2S      ;YES, DO IT.
2750 014456 000025      MICPC=MICPC+1

```

3S:

2748	014512	100403	WORD	.S	
2749	014514		MOVE	SPAD <4>,BREG	
2750		000026	MICPC=MICPC+1		
2751	014514	060604	.WORD	.S	
2752	014516		SHFBRT		;NO, CONTINUE...
2753		000027	MICPC=MICPC+1		
2754	014516	061620	.WORD	SBR!.SELB!.DGRSH	
2755	014520		BRT	4S	;IS IT DONE?...
2756		000030	MICPC=MICPC+1		
2757	014520	103432	.WORD	.S	
2758	014522		SBR	2S	;NO, CONTINUE...
2759		000031	MICPC=MICPC+1		
2760	014522	100403	.WORD	.S	
2761	014524		4S:		
2762	014524		SFLOT	OUT1,INP1,6,0,0,11S,12S,13S,14S	
2763	014524		11S:	MOVE # 177,BREG ;START WITH BIT 7.	
2764		000032	MICPC=MICPC+1		
2765	014524	000577	.WORD	.S	
2766	014526		12S:		
2767	014526		MOVE	BREG,OUT1 <6> ;SET THE BIT.	
2768		000033	MICPC=MICPC+1		
2769	014526	061226	.WORD	.S	
2770	014530		MOVE	INP1 <6>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.	
2771		000034	MICPC=MICPC+1		
2772	014530	123140	.WORD	.S	
2773	014532		MOVE	BREG,SPAD <4>	
2774		000035	MICPC=MICPC+1		
2775	014532	063224	.WORD	.S	
2776	014534		\$IFEQ	BREG,SPAD <0> 13S ;CHECK THE DATA...	
2777					
2778					
2779	014534		SUB2C	SPAD <0>,BREG,NOP	
2780		000036	MICPC=MICPC+1		
2781	014534	060360	.WORD	.S	
2782	014536		BZ	13S	
2783		000037	MICPC=MICPC+1		
2784	014536	101452	.WORD	.S	
2785	014540		MOVE	BREG,OUT1 <CSR4> ;GOOD DATA...	
2786		000040	MICPC=MICPC+1		
2787	014540	061224	.WORD	.S	
2788	014542		MOVE	INP1 <6>,OUT1 <CSR5>;BAD DATA...	
2789		000041	MICPC=MICPC+1		
2790	014542	121145	.WORD	.S	
2791	014544		MOVE	# 1,MEM ;TYPE OF ERROR...	
2792		000042	MICPC=MICPC+1		
2793	014544	002401	.WORD	.S	
2794	014546		MOVE	MEM,OUT1 <CSR3> ;	
2795		000043	MICPC=MICPC+1		
2796	014546	041223	.WORD	.S	
2797	014550		MOVE	# 6,MEM ;	
2798		000044	MICPC=MICPC+1		
2799	014550	002406	.WORD	.S	
2800	014552		MOVE	MEM,OUT1 <CSR7> ;REG. ADDRESS.	
2801		000045	MICPC=MICPC+1		
2802	014552	041227	.WORD	.S	
2803	014554		CALL	EROR ;REPORT DATA ERROR.	

2804 014554
2805 014554
2806 014554
2807 014556
2808 014556
2809 014556
2810 014560
2811 014560
2812 014560
2813 014562
2814 014562
2815 014564
2816 014564
2817 014564
2818 014564
2819 014564
2820 014566
2821 014566
2822 014570
2823 014570
2824 014572
2825 014572
2826 014574
2827 014574
2828 014576
2829 014576
2830 014600
2831 014600
2832 014602
2833 014602
2834 014602
2835 014604
2836 014604
2837 014606
2838 014606
2839 014610
2840 014610
2841 014610
2842 014610
2843 014610
2844 014610
2845 014610
2846 014610
2847 014610
2848 014610
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       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.
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.
SFBRT
MICPC=MICPC+1
.WORD     .SBR!.SELB!.DBRSH
BB7       125
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.
SFLT1    OUT1,INP1,7,0
SXZ

```

```

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

```

KPC11 MICRO PROCESSOR IBUS# TESTS

mlt

```

2850 014610          STSTM
          ; TEST 4
          -----
014610 012737 000004 001202 TST4:  MOV  #4,STSTM          ; LOAD THE NO. OF THIS TEST
014616 012737 015014 001442      MOV  #ST5,NEXT        ; POINT TO THE START OF NEXT TEST.
          ;R1 CONTAINS BASE KPC11 ADDRESS
          ;LOAD-VERIFY-WAIT.
014624 004737 035536          JSR  PC,LDVINT
014630 014644          MCT4
014632 104022          ERROR 22          ; TIME OUT ERROR...
014634 012706 001200      MOV  #STACK,SP      ; RESET STACK...
014640 000177 164576          JMP  #NEXT          ; GO TO NEXT TEST...
014644          MCT4:
014644 21S:
          MOVE  #0,BREG          ; SET TO CLEAR SPAD 16
          MICPC=MICPC+1
          .WORD  .S.
014644 000000          .MOVE  BREG,SPAD <16> ; FOR RETURN ADDRESS PURPOSES...
014646 000400          .WORD  .S.
          MICPC=MICPC+1
          .WORD  .S.
014646 000001          SFLOT OUT1,INP1,7,0,1,1S,2S,3S,4S
014650 063236          1S: MOVE  #200,BREG ; START WITH BIT 7.
          MICPC=MICPC+1
          .WORD  .S.
014650 000002          2S:
014652 000600          MOVE  BREG,OUT1 <7> ; SET THE BIT.
          MICPC=MICPC+1
          .WORD  .S.
014652 061227          .MOVE  INP1 <7>,SPAD <0> ; GET THE "FOUND" IN SCRATCH PAD.
014654 000004          MICPC=MICPC+1
          .WORD  .S.
014654 123160          .MOVE  BREG,SPAD <4>
014656 000005          MICPC=MICPC+1
014656 063224          $IFEQ  BREG,SPAD <0> 3S ; CHECK THE DATA...
          .WORD  .S.
014660          SUB2C  SPAD <0>,BREG,NOP
014660 000006          MICPC=MICPC+1
014660 060360          .WORD  .S.
014662          BZ 3S
          MICPC=MICPC+1
014662 000007          .WORD  .S.
014664 101422          MOVE  BREG,OUT1 <CSR4> ; GOOD DATA...
          MICPC=MICPC+1
          .WORD  .S.
014664 061224          MOVE  INP1 <7>,OUT1 <CSRS>;BAD DATA...
          MICPC=MICPC+1
          .WORD  .S.
014666 000011          MOVE  #1,MEM          ; TYPE OF ERROR...
          MICPC=MICPC+1
          .WORD  .S.
014670 000012          MOVE  MEM,OUT1 <CSR3> ;
          MICPC=MICPC+1
          .WORD  .S.
014672 041223          MOVE  #7,MEM          ;
          MICPC=MICPC+1
014674 000014

```

```

2916 014674 002407 .WORD .S.
2917 014676 000015 MOVE MEM,OUT1 <CSR7> ;REG. ADDRESS.
2918 000015 MICPC=MICPC+1
2919 014676 041227 .WORD .S.
2920 014700 000016 CALL EROR ;REPORT DATA ERROR.
2921 014700 000016 MOVE # <MICPC+3>,BREG
2922 014700 000420 MICPC=MICPC+1
2923 014702 000017 .WORD .S.
2924 014702 104400 SBR EROR
2925 014702 000017 MICPC=MICPC+1
2926 014702 104400 .WORD .S.
2927 014704 000020 MOVE SPAD <4>,BREG ;RESTORE BREG...
2928 014704 000020 MICPC=MICPC+1
2929 014704 060604 .WORD .S.
2930 014706 000021 SBR 25 ;LOOP ON ERROR...
2931 014706 000021 MICPC=MICPC+1
2932 014706 100403 .WORD .S.
2933 014710 35: CALL SCP1 ;IS LOOP DATA SET.
2934 014710 000022 MOVE # <MICPC+3>,BREG
2935 014710 000022 MICPC=MICPC+1
2936 014710 000424 .WORD .S.
2937 014712 000023 SBR SCP1
2938 014712 000023 MICPC=MICPC+1
2939 014712 104427 .WORD .S.
2940 014714 000024 MOVE SPAD <4>,BREG
2941 014714 060604 MICPC=MICPC+1
2942 014714 060604 .WORD .S.
2943 014716 000025 SBR 25 ;YES, DO IT.
2944 014716 000025 MICPC=MICPC+1
2945 014716 100403 .WORD .S.
2946 014720 000026 MOVE SPAD <4>,BREG
2947 014720 060604 MICPC=MICPC+1
2948 014720 060604 .WORD .S.
2949 014722 000027 SHFBRT ;NO, CONTINUE...
2950 014722 061620 MICPC=MICPC+1
2951 014722 000030 .WORD .SBR!..SELB!..DBRSH
2952 014724 000030 BBT 45 ;IS IT DONE?...
2953 014724 103432 MICPC=MICPC+1
2954 014726 000031 .WORD .S.
2955 014726 100403 SBR 25 ;NO, CONTINUE...
2956 014726 000031 MICPC=MICPC+1
2957 014726 100403 .WORD .S.
2958 014730 45:
2959 014730 SFLOT OUT1,IMP1,7,0,0,115,125,135,145
2960 014730 115: MOVE # 177,BREG ;START WITH BIT 7.
2961 014730 000032 MICPC=MICPC+1
2962 014730 000577 .WORD .S.
2963 014732 125:
2964 014732 MOVE BREG,OUT1 <7> ;SET THE BIT.
2965 014732 000033 MICPC=MICPC+1
2966 014732 051227 .WORD .S.
2967 014734 MOVE IMP1 <7>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
2968 014734 000034 MICPC=MICPC+1
2969 014734 123160 .WORD .S.
2970 014736 MOVE BREG,SPAD <4>
2971 000035 MICPC=MICPC+1

```

```

3022 014736 063224 .WORD S
3023 014740 BREG,SPAD <0> 135 ;CHECK THE DATA...
3024 014740 SUBEC SPAD <0>,BREG,NOP
3025 000036 MICPC=MICPC+1
3026 014740 060360 .WORD S
3027 014742 BZ 135
3028 000037 MICPC=MICPC+1
3029 014742 101482 .WORD S
3030 014744 MOVE BREG,OUT1 <CSR4> ;GOOD DATA...
3031 000040 MICPC=MICPC+1
3032 014746 061224 MOVE INP1 <?>,OUT1 <CSR5>;BAD DATA...
3033 000041 MICPC=MICPC+1
3034 014746 121166 .WORD S
3035 014750 MOVE B 1, MEM ;TYPE OF ERROR...
3036 000042 MICPC=MICPC+1
3037 014750 002401 .WORD S
3038 014752 MOVE MEM,OUT1 <CSR3> ;
3039 000043 MICPC=MICPC+1
3040 014752 041223 .WORD S
3041 014754 MOVE B 7, MEM ;
3042 000044 MICPC=MICPC+1
3043 014754 002407 .WORD S
3044 014756 MOVE MEM,OUT1 <CSR7> ;REG. ADDRESS.
3045 000045 MICPC=MICPC+1
3046 014756 041227 .WORD S
3047 014760 CALL ENOR ;REPORT DATA ERROR.
3048 014760 MOVE B <MICPC+3>,BREG
3049 000046 MICPC=MICPC+1
3050 014760 000450 .WORD S
3051 014762 SER ENOR
3052 000047 MICPC=MICPC+1
3053 014762 104400 .WORD S
3054 014764 MOVE SPAD <4>,BREG ;RESTORE BREG...
3055 000050 MICPC=MICPC+1
3056 014764 060604 .WORD S
3057 014766 SER 125 ;LOOP ON ERROR...
3058 000051 MICPC=MICPC+1
3059 014766 100433 .WORD S
3060 014770 135: CALL SCP1 ;IS LOOP DATA SET.
3061 014770 MOVE B <MICPC+3>,BREG
3062 000052 MICPC=MICPC+1
3063 014770 000454 .WORD S
3064 014772 SER SCP1
3065 000053 MICPC=MICPC+1
3066 014772 104427 .WORD S
3067 014774 MOVE SPAD <4>,BREG
3068 000054 MICPC=MICPC+1
3069 014774 060604 .WORD S
3070 014776 SER 125 ;YES, DO IT.
3071 000055 MICPC=MICPC+1
3072 014776 100433 .WORD S
3073 015000 MOVE SPAD <4>,BREG
3074 000056 MICPC=MICPC+1

```

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

KMC11 MICRO PROCESSOR IBUS* TESTS

```

3028 015000 060604 .WORD .S.
3029 015002 SHFBRT ;NO, CONTINUE...
3030 000057 MICPC=MICPC+1
3031 015002 061620 .WORD SBR!..SELB!..DBRSH
3032 015004 BBT 128 ;
3033 000060 MICPC=MICPC+1
3034 015004 103433 .WORD .S.
145:
3035 015006 CALL SCPE
3036 015006 MOVE # (MICPC+3),BREG
3037 015006 MICPC=MICPC+1
3038 000061 .WORD .S.
3039 015006 000463 SBR SCPE
3040 015010 MICPC=MICPC+1
3041 000062 .WORD .S.
3042 015010 104454 SBR 21$
3043 015012 MICPC=MICPC+1
3044 000063 .WORD .S.
3045 015012 100400 $FLT1 OUT1,INP1,10,236
3046 015014 SXZ
3047 015014
3048
3049
3050 ;***** TEST 5 *****
3051 ;* MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
3052 ;* FLOAT A 1 THROUGH REGISTER OUT1 <10>
3053 ;* FLOAT A 0 THROUGH REGISTER OUT1 <10>
3054 ;* THE NPR RD BIT (BIT0) IS MASKED DURING THIS TEST.
3055 015014 SXZ
3056 ;*****
3057
3058 015014 $STSN
3059 ; TEST 5
3060
3061 015014 012737 000005 001202 TST5: MOV #5,$STSNM ; LOAD THE NO. OF THIS TEST
3062 015022 012737 015234 001442 MOV #TST6,NEXT ; POINT TO THE START OF NEXT TEST.
3063 ;R1 CONTAINS BASE KMC11 ADDRESS
3064 015030 004737 035536 JSR PC,LDVRMT ;LOAD-VERIFY-WAIT.
3065 015034 015050 MCTS
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...
3069 015050 MCTS:
3070 015050 21$:
3071 015050 MOVE #0,BREG ;SET TO CLEAR SPAD 16
3072 000000 MICPC=MICPC+1
3073 015050 000400 .WORD .S.
3074 015052 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES...
3075 000001 MICPC=MICPC+1
3076 015052 063236 .WORD .S.
3077 015054 $FLOT OUT1,INP1,10,236,1,15,25,35,45
3078 015054 1$: MOVE #200,BREG ;START WITH BIT 7.
3079 000002 MICPC=MICPC+1
3080 015054 000600 .WORD .S.
3081 015056 2$:
3082 015056 MOVE BREG,SPAD <4> ;SAVE BREGISTER.
3083 000003 MICPC=MICPC+1

```

KMC11 MICRO PROCESSOR IBUS* TESTS

```

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

3140 015122
 3141 015122
 3142 000025
 3143 015122 000427
 3144 015124
 3145 000026
 3146 015124 104427
 3147 015126
 3148 000027
 3149 015126 060604
 3150 015130
 3151 000030
 3152 015130 100403
 3153 015132
 3154 000031
 3155 015132 060604
 3156 015134
 3157 000032
 3158 015134 061620
 3159 015136
 3160 000033
 3161 015136 103435
 3162 015140
 3163 000034
 3164 015140 100403
 3165 015142
 3166 015142
 3167 015142
 3168 000035
 3169 015142 000577
 3170 015144
 3171 015144
 3172 000036
 3173 015144 063224
 3174 015146
 3175 000037
 3176 015146 002636
 3177 015150
 3178 000040
 3179 015150 043221
 3180 015152
 3181 000041
 3182 015152 060661
 3183 015154
 3184 000042
 3185 015154 061230
 3186 015156
 3187 000043
 3188 015156 123200
 3189 015160
 3190
 3191
 3192 015160
 3193 000044
 3194 015160 060360
 3195 015162

```

3$: CALL SCP1 ;IS LOOP DATA SET.
MOVE B (<MICPC+3>),BREG
MICPC=MICPC+1
.WORD .S.
SBR SCP1
MICPC=MICPC+1
.WORD .S.
MOVE SPAD <4>,BREG
MICPC=MICPC+1
.WORD .S.
SBR 2$ ;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 4$ ;IS IT DONE?...
MICPC=MICPC+1
.WORD .S.
SBR 2$ ;NO, CONTINUE...
MICPC=MICPC+1
.WORD .S.
4$: SFLOT OUT1,INP1,10,236,0,11$,12$,13$,14$
11$: MOVE B 177,BREG ;START WITH BIT 7.
MICPC=MICPC+1
.WORD .S.
12$: MOVE BREG,SPAD <4> ;SAVE BREGISTER.
MICPC=MICPC+1
.WORD .S.
MOVE B 236,MEM ;CLEAR THE UNWANTED BITS
MICPC=MICPC+1
.WORD .S.
MOVE MEM,SPAD <1> ;CLEAR THE UNWANTED BITS.
MICPC=MICPC+1
.WORD .S.
AND SPAD <1>,BREG,BREG ;CLEAR THE UNWANTED BITS.
MICPC=MICPC+1
.WORD .S.
MOVE BREG,OUT1 <10> ;SET THE BIT.
MICPC=MICPC+1
.WORD .S.
MOVE INP1 <10>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
MICPC=MICPC+1
.WORD .S.
SIFEQ BREG,SPAD <0> 13$ ;CHECK THE DATA...

SUB2C SPAD <0>,BREG,NOP
MICPC=MICPC+1
.WORD .S.
BZ 13$
    
```

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 <CSR5>;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 IBJS# TESTS

```

3308 015304 000006 AND SPAD <1>,BREG,BREG ;CLEAR THE UNWANTED BITS.
3309 015304 060661 MICPC=MICPC+1
3310 015304 060661 .WORD .S.
3311 015306 MOVE BREG,OUT1 <11> ;SET THE BIT.
3312 015306 000007 MICPC=MICPC+1
3313 015306 061231 .WORD .S.
3314 015310 MOVE INP1 <11>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
3315 015310 000010 MICPC=MICPC+1
3316 015310 123220 .WORD .S.
3317 015312 MOVE # 20, MEM ;SET TO SET PGM.CLK BIT.
3318 015312 000011 MICPC=MICPC+1
3319 015312 002420 .WORD .S.
3320 015314 MOVE MEM,SPAD <1> ; " " " " " "
3321 015314 000012 MICPC=MICPC+1
3322 015314 043221 .WORD .S.
3323 015316 OR SPAD <1>,BREG,BREG ;SET PGM.CLK. BIT IN BREG.
3324 015316 000013 MICPC=MICPC+1
3325 015316 060701 .WORD .S.
3326 015320 SIFE0 BREG,SPAD <0> 3S ;CHECK THE DATA...
3327 015320 SUB2C SPAD <0>,BREG,NOP
3328 015320 000014 MICPC=MICPC+1
3329 015320 060360 .WORD .S.
3330 015322 BZ 3S
3331 015322 000015 MICPC=MICPC+1
3332 015322 101430 .WORD .S.
3333 015324 MOVE BREG,OUT1 <CSR4> ;GOOD DATA...
3334 015324 000016 MICPC=MICPC+1
3335 015324 061224 .WORD .S.
3336 015326 MOVE INP1 <11>,OUT1 <CSRS>;BAD DATA...
3337 015326 000017 MICPC=MICPC+1
3338 015326 121225 .WORD .S.
3339 015330 MOVE # 1, MEM ;TYPE OF ERROR...
3340 015330 000020 MICPC=MICPC+1
3341 015330 002401 .WORD .S.
3342 015332 MOVE MEM,OUT1 <CSR3> ;
3343 015332 000021 MICPC=MICPC+1
3344 015332 041223 .WORD .S.
3345 015334 MOVE # 11, MEM ;
3346 015334 000022 MICPC=MICPC+1
3347 015334 002411 .WORD .S.
3348 015336 MOVE MEM,OUT1 <CSR7> ;REG. ADDRESS.
3349 015336 000023 MICPC=MICPC+1
3350 015336 041227 .WORD .S.
3351 015340 CALL EROR ;REPORT DATA ERROR.
3352 015340 000024 MICPC=MICPC+1
3353 015340 000426 .WORD .S.
3354 015342 SBR EROR
3355 015342 000025 MICPC=MICPC+1
3356 015342 104400 .WORD .S.
3357 015344 MOVE SPAD <4>,BREG ;RESTORE BREG...
3358 015344 000026 MICPC=MICPC+1
3359 015344 060604 .WORD .S.
3360 015346 SBR 2S ;LOOP ON ERROR...

```

1-2

KNC11 MICRO PROCESSOR IBUS* TESTS

3364 000027
 3365 015346 100403
 3366 015350
 3367 015350
 3368 000030
 3369 015350 000432
 3370 015352
 3371 000031
 3372 015352 104427
 3373 015354
 3374 000032
 3375 015354 060604
 3376 015356
 3377 000033
 3378 015356 100403
 3379 015360
 3380 000034
 3381 015360 060604
 3382 015362
 3383 000035
 3384 015362 061620
 3385 015364
 3386 000036
 3387 015364 103440
 3388 015366
 3389 000037
 3390 015366 100403
 3391 015370
 3392 015370
 3393 015370
 3394 000040
 3395 015370 000577
 3396 015372
 3397 000041
 3398 015372 063224
 3399 015374
 3400 000042
 3401 015374 002415
 3402 015376
 3403 000043
 3404 015376 043221
 3405 015400
 3406 000044
 3407 015400 060661
 3408 015402
 3409 000045
 3410 015402 061231
 3411 015404
 3412 000046
 3413 015404 123220
 3414 015406
 3415 000047
 3416 015406 002420
 3417 015410
 3418 015410
 3419 000050

```

MICPC=MICPC+1
.WORD .S
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
.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: SFLOT OUT1,INP1,11,015,0,115,125,135,145
115: MOVE # 177,BREG ;START WITH BIT 7.
MICPC=MICPC+1
.WORD .S
125: 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
AND SPAD <1>,BREG,BREG ;CLEAR THE UNWANTED BITS.
MICPC=MICPC+1
.WORD .S
MOVE BREG,OUT1 <11> ;SET THE BIT.
MICPC=MICPC+1
.WORD .S
MOVE INP1 <11>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
MICPC=MICPC+1
.WORD .S
MOVE # 20,MEM ;SET TO SET PGH.CLK BIT.
MICPC=MICPC+1
.WORD .S
MOVE MEM,SPAD <1> ; " " " " " "
MICPC=MICPC+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 SIFEG 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 MICPC=MICPC+1
015434 000464 .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.
138: 015444 MOVE # <MICPC+3>,BREG
000066 MICPC=MICPC+1
015444 000470 .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
    
```

KMC11 MICRO PROCESSOR IBUS* TESTS

```

3175 015452 100441
3176 015452 000072
3177 015452 060604
3178 015452 000073
3179 015456 061620
3180 015460 000074
3181 015460 103441
3182 015460
3183 015460
3184 015462 000075
3185 015462 000477
3186 015464 000076
3187 015464 104454
3188 015466 000077
3189 015466 100400
3190 015470
3191 015470
3192 015470
3193 015470
3194 015470
3195 015470
3196 015470
3197 015470
3198 015470
3199 015470
3200 015470
3201 015470
3202 015470
3203 015470
3204 015470
3205 015470
3206 015470
3207 015470
3208 015470
3209 015470
3210 015470
3211 015470 012737 000007 001202
3212 015476 012737 015674 001442
3213 015504 004737 035536
3214 015510 015524
3215 015512 104022
3216 015514 012706 001200
3217 015520 000177 163716
3218 015524
3219 015524
3220 015524
3221 015524
3222 015524
3223 015524
3224 015524
3225 015524
3226 015524
3227 015524
3228 015524
3229 015524
3230 015524
3231 015524
3232 015524
3233 015524
3234 015524
3235 015524
3236 015524
3237 015524
3238 015524
3239 015524
3240 015524
3241 015524
3242 015524
3243 015524
3244 015524
3245 015524
3246 015524
3247 015524
3248 015524
3249 015524
3250 015524
3251 015524
3252 015524
3253 015524
3254 015524
3255 015524
3256 015524
3257 015524
3258 015524
3259 015524
3260 015524
3261 015524
3262 015524
3263 015524
3264 015524
3265 015524
3266 015524
3267 015524
3268 015524
3269 015524
3270 015524
3271 015524
3272 015524
3273 015524
3274 015524
3275 015524
3276 015524
3277 015524
3278 015524
3279 015524
3280 015524
3281 015524
3282 015524
3283 015524
3284 015524
3285 015524
3286 015524
3287 015524
3288 015524
3289 015524
3290 015524
3291 015524
3292 015524
3293 015524
3294 015524
3295 015524
3296 015524
3297 015524
3298 015524
3299 015524
3300 015524
3301 015524
3302 015524
3303 015524
3304 015524
3305 015524
3306 015524
3307 015524
3308 015524
3309 015524
3310 015524
3311 015524
3312 015524
3313 015524
3314 015524
3315 015524
3316 015524
3317 015524
3318 015524
3319 015524
3320 015524
3321 015524
3322 015524
3323 015524
3324 015524
3325 015524
3326 015524
3327 015524
3328 015524
3329 015524
3330 015524
3331 015524
3332 015524
3333 015524
3334 015524
3335 015524
3336 015524
3337 015524
3338 015524
3339 015524
3340 015524
3341 015524
3342 015524
3343 015524
3344 015524
3345 015524
3346 015524
3347 015524
3348 015524
3349 015524
3350 015524
3351 015524

```

```

        .WORD      $
        MOVE      SPAD <4>,BREG
        MICPC=MICPC+1
        .WORD      .S.
        SHFBRT
        ;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      21$
        MICPC=MICPC+1
        .WORD      .S.
$FLT1
        OUTO,INPO,0,0
$XZ
;***** TEST 7 *****
;* MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
;* FLOAT A 1 THROUGH REGISTER OUTO <0>
;* FLOAT A 0THROUGH REGISTER OUTO <0>
$XZ
;*****
$STSN
; TEST 7
-----
TST7:  MOV      #7,$STSNM
        MOV      $TST10,NEXT
        ; LOAD THE NO. OF THIS TEST
        ; POINT TO THE START OF NEXT TEST.
        ;R1 CONTAINS BASE KMC11 ADDRESS
        ;LOAD-VERIFY-WAIT.
        JSR      PC,LDRVWT
        MCT7
        ERROR    22
        ;TIME OUT ERROR...
        MOV      $STACK,SP
        ;RESET STACK...
        JMP      $NEXT
        ;GO TO NEXT TEST...
MCT7:  21$:
        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.
$FLOT
        OUTO,INPO,0,0,1,1$,2$,3$,4$
1$: MOVE      # 200,BREG
        ;START WITH BIT 7.
        MICPC=MICPC+1
        .WORD      .S.
2$:

```

3560	015532	000003	MOVE BREG,OUT0 <0> ;SET THE BIT.
3561	015532	062220	NICPC=NICPC+1
3562	015534	000004	.WORD \$
3563	015534	023000	MOVE INPD <0>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
3564	015536	000005	NICPC=NICPC+1
3565	015536	063224	.WORD \$
3566	015540		MOVE BREG,SPAD <4>
3567			NICPC=NICPC+1
3568			.WORD \$
3569			SIFEQ BREG,SPAD <0> 3\$;CHECK THE DATA...
3570	015540	000006	SUBC SPAD <0>,BREG,NOP
3571	015540	060360	NICPC=NICPC+1
3572	015542		.WORD \$
3573	015542	000007	BZ 3\$
3574	015542	101422	NICPC=NICPC+1
3575	015544		.WORD \$
3576	015544	000010	MOVE BREG,OUT1 <CSR4> ;GOOD DATA...
3577	015544	061224	NICPC=NICPC+1
3578	015546		.WORD \$
3579	015546	000011	MOVE INPD <0>,OUT1 <CSR5>;BAD DATA...
3580	015550	021005	NICPC=NICPC+1
3581	015550		.WORD \$
3582	015550	000012	MOVE #1,MEM ;TYPE OF ERROR...
3583	015550	002401	NICPC=NICPC+1
3584	015552		.WORD \$
3585	015552	000013	MOVE MEM,OUT1 <CSR3> ;
3586	015554	041223	NICPC=NICPC+1
3587	015554		.WORD \$
3588	015554	000014	MOVE #0,MEM ;
3589	015554	002400	NICPC=NICPC+1
3590	015556		.WORD \$
3591	015556	000015	MOVE MEM,OUT1 <CSR7> ;REG. ADDRESS.
3592	015560	041227	NICPC=NICPC+1
3593	015560		.WORD \$
3594	015560		CALL EROR ;REPORT DATA ERROR.
3595	015560		MOVE # <NICPC+3>,BREG
3596	015560	000016	NICPC=NICPC+1
3597	015560	000420	.WORD \$
3598	015562		SBR EROR
3599	015562	000017	NICPC=NICPC+1
3600	015562	104400	.WORD \$
3601	015564		MOVE SPAD <4>,BREG ;RESTORE BREG...
3602	015564	000020	NICPC=NICPC+1
3603	015566	060604	.WORD \$
3604	015566		SBR 2\$;LOOP ON ERROR...
3605	015566	000021	NICPC=NICPC+1
3606	015566	100403	.WORD \$
3607	015570		CALL SCP1 ;IS LOOP DATA SET.
3608	015570		MOVE # <NICPC+3>,BREG
3609	015570	000022	NICPC=NICPC+1
3610	015570	000424	.WORD \$
3611	015572		SBR SCP1
3612	015572	000023	NICPC=NICPC+1
3613	015572	104427	.WORD \$

3\$:

KMC11 MICRO PROCESSOR IBUS TESTS

```

3588 015574          MOVE   SPAD <4>,BREG
3589          NICPC=NICPC+1
3590 015574 000024   .WORD  .S.
3591 015576 060604   SBR    25          ;YES, DO IT.
3592          NICPC=NICPC+1
3593 015576 000025   .WORD  .S.
3594 015600 100403   MOVE   SPAD <4>,BREG
3595          NICPC=NICPC+1
3596 015600 000026   .WORD  .S.
3597 015602 060604   SHFBT          ;NO, CONTINUE...
3598          NICPC=NICPC+1
3599 015602 000027   .WORD  .SBR!..SELB!..DBRSH
3600 015604 061620   BZ     45          ;IS IT DONE?...
3601          NICPC=NICPC+1
3602 015604 000030   .WORD  .S.
3603 015606 103432   SBR    25          ;NO, CONTINUE...
3604          NICPC=NICPC+1
3605 015606 000031   .WORD  .S.
3606 015610          45:
3607 015610 SFLOT  OUT0,INPO,0,0,0,115,125,135,145
3608 015610 115:  MOVE  # 177,BREG ;START WITH BIT 7.
3609          NICPC=NICPC+1
3610 015610 000032   .WORD  .S.
3611 015612          6:
3612 015612          MOVE  BREG,OUT0 <0> ;SET THE BIT.
3613          NICPC=NICPC+1
3614 015612 000033   .WORD  .S.
3615 015614 062220   MOVE  INPO <0>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
3616          NICPC=NICPC+1
3617 015614 000034   .WORD  .S.
3618 015616 023000   MOVE  BREG,SPAD <4>
3619          NICPC=NICPC+1
3620 015616 000035   .WORD  .S.
3621 015620 063224   SIFEQ  BREG,SPAD <0> 135 ;CHECK THE DATA...
3622          NICPC=NICPC+1
3623 015620          SUB2C  SPAD <0>,BREG,NOP
3624          NICPC=NICPC+1
3625 015620 000036   .WORD  .S.
3626 015622 060360   BZ     135
3627          NICPC=NICPC+1
3628 015622 000037   .WORD  .S.
3629 015624 101452   MOVE  BREG,OUT1 <CSR4> ;GOOD DATA...
3630          NICPC=NICPC+1
3631 015624 000040   .WORD  .S.
3632 015626 061224   MOVE  INPO <0>,OUT1 <CSR5>;BAD DATA...
3633          NICPC=NICPC+1
3634 015626 000041   .WORD  .S.
3635 015630 021005   MOVE  # 1, MEM          ;TYPE OF ERROR...
3636          NICPC=NICPC+1
3637 015630 000042   .WORD  .S.
3638 015632 002401   MOVE  MEM,OUT1 <CSR3> ;
3639          NICPC=NICPC+1
3640 015632 000043   .WORD  .S.
3641 015634 041223   MOVE  # 0, MEM          ;
3642          NICPC=NICPC+1
3643 015634 000044   .WORD  .S.

```

3644 015634 002400
 3645 015636 000045
 3646 015636 041227
 3647 015640 000046
 3648 015640 000450
 3649 015640 000047
 3650 015642 104400
 3651 015644 000050
 3652 015644 060604
 3653 015646 000051
 3654 015646 100433
 3655 015650 000052
 3656 015652 000454
 3657 015652 000053
 3658 015654 104427
 3659 015654 000054
 3660 015654 060604
 3661 015656 000055
 3662 015660 100433
 3663 015660 000056
 3664 015660 060604
 3665 015662 000057
 3666 015662 061620
 3667 015664 000060
 3668 015664 103433
 3669 015666 000061
 3670 015670 000463
 3671 015670 000062
 3672 015672 104454
 3673 015674 000063
 3674 015674 100400
 3675 015674
 3676
 3677
 3678
 3679
 3680
 3681
 3682
 3683
 3684
 3685
 3686
 3687
 3688
 3689
 3690
 3691
 3692
 3693
 3694
 3695
 3696
 3697
 3698
 3699

```

.WORD .S.
MOVE MEM,OUT1 (CSR7) ;REG. ADDRESS.
MICPC=MICPC+1
.WORD .S.
CALL ERROR ;REPORT DATA ERROR.
MOVE # (MICPC+3),BREG
MICPC=MICPC+1
.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
13$: .WORD .S.
CALL SCP1 ;IS LOOP DATA SET.
MOVE # (MICPC+3),BREG
MICPC=MICPC+1
.WORD .S.
SBR SCP1
MICPC=MICPC+1
.WORD .S.
MOVE SPAD (4),BREG
MICPC=MICPC+1
.WORD .S.
SBR 128 ;YES, DO IT.
MICPC=MICPC+1
.WORD .S.
MOVE SPAD (4),BREG
MICPC=MICPC+1
.WORD .S.
SHFBRT ;NO, CONTINUE...
MICPC=MICPC+1
.WORD .SBR!.SELB!.DBRSH
BB7 128 ;
MICPC=MICPC+1
14$: .WORD .S.
CALL SCPE
MOVE # (MICPC+3),BREG
MICPC=MICPC+1
.WORD .S.
SBR SCPE
MICPC=MICPC+1
.WORD .S.
SBR 218
MICPC=MICPC+1
.WORD .S.
SFLT1 OUTO,INPO,1,0
SXZ

```

***** 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 #STSTN1,NEXT ; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
;LOAD-VERIFY-WAIT.
3710
3711 015710 004737 035536 JSR PC,LDVWRT
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
3717 015730 MCT10:
3718 015730 21$: MOVE #0,BREG ;SET TO CLEAR SPAD 16
3719 000000 MICPC=MICPC+1
3720 015730 000400 .WORD .S.
3721 015732 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES...
3722 000001 MICPC=MICPC+1
3723 015732 063236 .WORD .S.
3724 015734 SFLOT OUT0,INPO,1,0,1,15,25,35,45
3725 015734 15: MOVE #200,BREG ;START WITH BIT 7.
3726 000002 MICPC=MICPC+1
3727 015734 000600 .WORD .S.
3728 015736 25:
3729 015736 MOVE BREG,OUT0 <1> ;SET THE BIT.
3730 000003 MICPC=MICPC+1
3731 015736 062221 .WORD .S.
3732 015740 MOVE INPO <1>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
3733 000004 MICPC=MICPC+1
3734 015740 023020 .WORD .S.
3735 015742 MOVE BREG,SPAD <4>
3736 000005 MICPC=MICPC+1
3737 015742 063224 .WORD .S.
3738 015744 SIFEQ BREG,SPAD <0> 35 ;CHECK THE DATA...
3739
3740
3741 015744 SUB2C SPAD <0>,BREG,NOP
3742 000006 MICPC=MICPC+1
3743 015744 060360 .WORD .S.
3744 015746 BZ 35
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 <CSRS>;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 015760 .WORD .S.
3759 015760 MOVE # 1, MEM ;
3760 000014 MICPC=MICPC+1
3761 015760 002401 .WORD .S.
3762 015762 MOVE MEM OUT1 (CSR7) ;REG. ADDRESS.
3763 000015 MICPC=MICPC+1
3764 015762 041227 .WORD .S.
3765 015764 CALL ERROR ;REPORT DATA ERROR.
3766 015764 MOVE # (MICPC+3), BREG
3767 000016 MICPC=MICPC+1
3768 015764 000420 .WORD .S.
3769 015766 SBR ERROR
3770 000017 MICPC=MICPC+1
3771 015766 104400 .WORD .S.
3772 015770 MOVE SPAD (4), BREG ;RESTORE BREG...
3773 000020 MICPC=MICPC+1
3774 015770 060604 .WORD .S.
3775 015772 SBR 25 ;LOOP ON ERROR...
3776 000021 MICPC=MICPC+1
3777 015772 100403 .WORD .S.
3778 015774 35: CAI SCP1 ;IS LOOP DATA SET.
3779 015774 MOVE # (MICPC+3), BREG
3780 000022 MICPC=MICPC+1
3781 015774 000424 .WORD .S.
3782 015776 SBR SCP1
3783 000023 MICPC=MICPC+1
3784 015776 104427 .WORD .S.
3785 016000 MOVE SPAD (4), BREG
3786 000024 MICPC=MICPC+1
3787 016000 060604 .WORD .S.
3788 016002 SBR 25 ;YES, DO IT.
3789 000025 MICPC=MICPC+1
3790 016002 100403 .WORD .S.
3791 016004 MOVE SPAD (4), BREG
3792 000026 MICPC=MICPC+1
3793 016004 060604 .WORD .S.
3794 016006 SHFBRT ;NO, CONTINUE...
3795 000027 MICPC=MICPC+1
3796 016006 061620 .WORD .SBR!.SELB!.DBRSH
3797 016010 BB7 45 ;IS IT DONE?...
3798 000030 MICPC=MICPC+1
3799 016010 103432 .WORD .S.
3800 016012 SBR 25 ;NO, CONTINUE...
3801 000031 MICPC=MICPC+1
3802 016012 100403 .WORD .S.
3803 016014 45: SFL0T OUT0, INP0, 1, 0, 0, 115, 125, 135, 145
3804 016014 115: MOVE # 177, BREG ;START WITH BIT 7.
3805 016014 MICPC=MICPC+1
3806 000032 .WORD .S.
3807 016014 000577
3808 016016 125: MOVE BREG OUT0 (1) ;SET THE BIT.
3809 016016 MICPC=MICPC+1
3810 000033 .WORD .S.
3811 016016 062221
    
```

```

3812 016020 MOVE INFO (1),SPAD (0) ;GET THE "FOUND" IN SCRATCH PAD.
3813 000034 MICPC=MICPC+1
3814 016020 .WORD .S.
3815 016022 MOVE BREG,SPAD (4)
3816 000035 MICPC=MICPC+1
3817 016022 .WORD .S.
3818 016024 SIFEQ BREG,SPAD (0) 13S ;CHECK THE DATA...
3819
3820
3821 016024 SUB2C SPAD (0),BREG,NOP
3822 000036 MICPC=MICPC+1
3823 016024 .WORD .S.
3824 016026 BZ 13S
3825 000037 MICPC=MICPC+1
3826 016026 .WORD .S.
3827 016030 MOVE BREG,OUT1 (CSR4) ;GOOD DATA...
3828 000040 MICPC=MICPC+1
3829 016030 .WORD .S.
3830 016032 MOVE INFO (1),OUT1 (CSRS);BAD DATA...
3831 000041 MICPC=MICPC+1
3832 016032 .WORD .S.
3833 016034 MOVE 0(1),MEM ;TYPE OF ERROR...
3834 000042 MICPC=MICPC+1
3835 016034 .WORD .S.
3836 016036 MOVE MEM,OUT1 (CSR3) ;
3837 000043 MICPC=MICPC+1
3838 016036 .WORD .S.
3839 016040 MOVE 0(1),MEM ;
3840 000044 MICPC=MICPC+1
3841 016040 .WORD .S.
3842 016042 MOVE MEM,OUT1 (CSR7) ;REG. ADDRESS.
3843 000045 MICPC=MICPC+1
3844 016042 .WORD .S.
3845 016044 CALL EROR ;REPORT DATA ERROR.
3846 016044 MOVE 0 (MICPC+3),BREG
3847 000046 MICPC=MICPC+1
3848 016044 .WORD .S.
3849 016046 SBR EROR
3850 000047 MICPC=MICPC+1
3851 016046 .WORD .S.
3852 016050 MOVE SPAD (4),BREG ;RESTORE BREG...
3853 000050 MICPC=MICPC+1
3854 016050 .WORD .S.
3855 016052 SBR 12S ;LOOP ON ERROR...
3856 000051 MICPC=MICPC+1
3857 016052 .WORD .S.
3858 016054 CALL SCP1 ;IS LOOP DATA SET.
3859 016054 MOVE 0 (MICPC+3),BREG
3860 000052 MICPC=MICPC+1
3861 016054 .WORD .S.
3862 016056 SBR SCP1
3863 000053 MICPC=MICPC+1
3864 016056 .WORD .S.
3865 016060 MOVE SPAD (4),BREG
3866 000054 MICPC=MICPC+1
3867 016060 .WORD .S.
    
```

13S:

4

```

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    # <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    OUT0,INP0,2,0
3893
3894
3895          ;***** TEST 11 *****
3896          ;* MICRO PROCESSOR OUT0 REGISTER WRITE/READ TEST.
3897          ;* FLOAT A 1 THROUGH REGISTER OUT0 <2>
3898          ;* FLOAT A 0 THROUGH REGISTER OUT0 <2>
3899 016100          SXZ
3900
3901          ;*****
3902 016100          STSTN
3903
3904          ; TEST 11
3905          ;-----
3906 016100 012737 000011 001202 TST11: MOV    #11,STSTNM          ; LOAD THE NO. OF THIS TEST
3907 016106 012737 016304 001442 MOV    #TST12,NEXT        ; POINT TO THE START OF NEXT TEST.
3908          004737 035536 JSR    PC,LDRVNT          ;R1 CONTAINS BASE KNC11 ADDRESS
3909 016120 016134          MCT11          ;LOAD-VERIFY-WAIT.
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          MCT11:
3914          21S:
3915          MOVE    # 0,BREG          ;SET TO CLEAR SPAD 16
3916          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          SFLT0  OUT0,INP0,2,0,1,1S,2S,3S,4S
3922 016140          1S: MOVE    # 200,BREG          ;START WITH BIT 7.
3923          MICPC=MICPC+1

```

KMC11 MICRO PROCESSOR IBUS TESTS

3924 016140 000600
 3925 016142
 3926 016142
 3927 000003
 3928 016142 062222
 3929 016144
 3930 000004
 3931 016144 023040
 3932 016146
 3933 000005
 3934 016146 063224
 3935 016150
 3936
 3937
 3938 016150
 3939 000006
 3940 016150 060360
 3941 016152
 3942 000007
 3943 016152 101422
 3944 016154
 3945 000010
 3946 016154 061224
 3947 016156
 3948 000011
 3949 016156 021045
 3950 016160
 3951 000012
 3952 016160 002401
 3953 016162
 3954 000013
 3955 016162 041223
 3956 016164
 3957 000014
 3958 016164 002402
 3959 016166
 3960 000015
 3961 016166 041227
 3962 016170
 3963 016170
 3964 000016
 3965 016170 000420
 3966 016172
 3967 000017
 3968 016172 104400
 3969 016174
 3970 000020
 3971 016174 060604
 3972 016176
 3973 000021
 3974 016176 100403
 3975 016200
 3976 016200
 3977 000022
 3978 016200 000424
 3979 016202

```

25:  .WORD  .S.
      MOVE  BREG,OUT0 <2> ;SET THE BIT.
      MICPC=MICPC+1
      .WORD  .S.
      MOVE  INPO <2>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
      MICPC=MICPC+1
      .WORD  .S.
      MOVE  BREG,SPAD <4>
      MICPC=MICPC+1
      .WORD  .S.
      $IFEQ  BREG,SPAD <0> 35 ;CHECK THE DATA...

      SUB2C  SPAD <0>,BREG,NOP
      MICPC=MICPC+1
      .WORD  .S.
      BZ  35
      MICPC=MICPC+1
      .WORD  .S.
      MOVE  BREG,OUT1 <CSR4> ;GOOD DATA...
      MICPC=MICPC+1
      .WORD  .S.
      MOVE  INPO <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.
      MOVE  # 2,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
      .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
      .WORD  .S.
      CALL  SCP1 ;IS LOOP DATA SET.
      MOVE  # <MICPC+3>,BREG
      MICPC=MICPC+1
      .WORD  .S.
      SBR  SCP1
    
```

35:

```

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

```

```

MICPC=MICPC+1
MOVE SPAD (4),BREG
MICPC=MICPC+1
WORD .S.
SBR 23 ;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 23 ;NO, CONTINUE...
MICPC=MICPC+1
WORD .S.
45:
SFLOT OUTO,INPO,2,0,0,115,125,135,145
115: MOVE #177,BREG ;START WITH BIT 7.
MICPC=MICPC+1
WORD .S.
125:
MOVE BREG,OUTO (2) ;SET THE BIT.
MICPC=MICPC+1
WORD .S.
MOVE INPO (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 INPO (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.

```

4036	016244	000044	MOVE # 2 MEM ;
4037			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		B87 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		SFLT1
4089	016304		SXZ
4090			
4091			

KMC11 MICRO PROCESSOR IBUS TESTS

```

***** TEST 12 *****
; MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
; FLOAT A 1 THROUGH REGISTER OUTO <3>
; FLOAT A 0 THROUGH REGISTER OUTO <3>
;*****

4092
4093
4094
4095
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 #ST13,NEXT ; POINT TO THE START OF NEXT TEST.
4104 ;R1 CONTAINS BASE KMC11 ADDRESS
4105 016320 004737 035536 JSR PC,LDRWMT ;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 MCT12:
4111 215:
4112 016340 MOVE #0,BREG ;SET TO CLEAR SPAD 16
4113 MICPC=MICPC+1
4114 016340 000400 .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 15: MOVE #200,BREG ;START WITH BIT 7.
4120 000002 MICPC=MICPC+1
4121 016344 000600 .WORD .S.
4122 016346 25:
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 $IFEQ BREG,SPAD <0> 3$ ;CHECK THE DATA...
4133
4134
4135 016354 SUB2C SPAD <0>,BREG,NOP
4136 000006 MICPC=MICPC+1
4137 016354 060360 .WORD .S.
4138 016356 BZ 3$
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 EROR	;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 EROR	;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 48	;IS IT DONE?...
4192		000030	MICPC=MICPC+1	
4193	016420	103432	.WORD .S.	
4194	016422		SBR EROR	;NO, CONTINUE...
4195		000031	MICPC=MICPC+1	
4196	016422	100403	.WORD .S.	
4197	016424			
4198	016424			
4199	016424			
4200		000032		
4201	016424	000577		
4202	016426			
4203	016426			

45: SFLOT OUT0,INP0,3,0,0,115,125,135,145
 115: MOVE # 177,BREG ;START WITH BIT 7.
 125: MOVE BREG,OUT0 <3> ;SET THE BIT.

4201		000033	MICPC=MICPC+1	
4202	016436	062223	.WORD .S	
4203	016430		MOVE INFO <3>,SPAD <0>	;GET THE "FOUND" IN SCRATCH PAD.
4204		000034	MICPC=MICPC+1	
4205	016430	023060	.WORD .S	
4206	016432		MOVE BREG,SPAD <4>	
4207		000035	MICPC=MICPC+1	
4208	016432	063224	.WORD .S	
4209	016434		\$IFEQ BREG,SPAD <0> 13S	;CHECK THE DATA...
4210				
4211	016434			
4212				
4213	016434		SUB2C SPAD <0>,BREG,NOP	
4214		000036	MICPC=MICPC+1	
4215	016434	060360	.WORD .S	
4216	016436		BZ 13S	
4217		000037	MICPC=MICPC+1	
4218	016436	101452	.WORD .S	
4219	016440		MOVE BREG,OUT1 <CSR4>	;GOOD DATA...
4220		000040	MICPC=MICPC+1	
4221	016440	061224	.WORD .S	
4222	016442		MOVE INFO <3>,OUT1 <CSR5>;BAD DATA...	
4223		000041	MICPC=MICPC+1	
4224	016442	021065	.WORD .S	
4225	016444		MOVE # 1,MEH	;TYPE OF ERROR...
4226		000042	MICPC=MICPC+1	
4227	016444	002401	.WORD .S	
4228	016446		MOVE MEH,OUT1 <CSR3>	;
4229		000043	MICPC=MICPC+1	
4230	016446	041223	.WORD .S	
4231	016450		MOVE # 3,MEH	;
4232		000044	MICPC=MICPC+1	
4233	016450	002403	.WORD .S	
4234	016452		MOVE MEH,OUT1 <CSR7>	;REG. ADDRESS.
4235		000045	MICPC=MICPC+1	
4236	016452	041227	.WORD .S	
4237	016454		CALL EROR	;REPORT DATA ERROR.
4238			MOVE # <MICPC+3>,BREG	
4239		000046	MICPC=MICPC+1	
4240	016454	000450	.WORD .S	
4241	016456		SBR EROR	
4242		000047	MICPC=MICPC+1	
4243	016456	104400	.WORD .S	
4244	016460		MOVE SPAD <4>,BREG	;RESTORE BREG...
4245		000050	MICPC=MICPC+1	
4246	016460	060604	.WORD .S	
4247	016462		SBR 12S	;LOOP ON ERROR...
4248		000051	MICPC=MICPC+1	
4249	016462	100433	.WORD .S	
4250	016464		CALL SCP1	;IS LOOP DATA SET.
4251	016464		MOVE # <MICPC+3>,BREG	
4252		000052	MICPC=MICPC+1	
4253	016464	000454	.WORD .S	
4254	016466		SBR SCP1	
4255		000053	MICPC=MICPC+1	
4256	016466	104427	.WORD .S	
4257	016470		MOVE SPAD <4>,BREG	

13S:

KMC11 MICRO PROCESSOR IBUS TESTS

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

000054
060604
000055
100433
000056
060604
000057
061620
000060
103433
000061
000463
000062
104454
000063
100400
016470
016472
016472
016474
016474
016476
016476
016500
016500
016502
016502
016502
016504
016504
016506
016510
016510
016510
016510
016510
016516
016524
016530
016532
016534
016540
016544
016544
016544
016544
016544
016544
016544
016546
016546
016550

MICPC=MICPC+1
.WORD .S.
SRR 129 ;YES, DO IT.
MICPC=MICPC+1
.WORD .S.
MOVE SPAD <4>,BREG
MICPC=MICPC+1
.WORD .S.
SHFBRT ;NO, CONTINUE...
MICPC=MICPC+1
.WORD SRR!..SELB!..DBRSH
SRR 129 ;
MICPC=MICPC+1
.WORD .S.
149:
CALL SCPE
MOVE # <MICPC+3>,BREG
MICPC=MICPC+1
.WORD .S.
SRR SCPE
MICPC=MICPC+1
.WORD .S.
SRR 218
MICPC=MICPC+1
.WORD .S.
SFLT1
OUTO,INPO,4,0
SXZ

;***** TEST 13 *****
;# MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
;# FLOAT A 1 THROUGH REGISTER OUTO <4>
;# FLOAT A 0 THROUGH REGISTER OUTO <4>
SXZ
;*****
STSTN
; TEST 13
TST13: MOV #13,STSTNM ; LOAD THE NO. OF THIS TEST
MOV #ST14,NEXT ; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
JSR PC,LDVRMT ;LOAD-VERIFY-WAIT.
MCT13
ERROR 22 ;TIME OUT ERROR...
MOV #STACK,SP ;RESET STACK...
JMP #NEXT ;GO TO NEXT TEST...
MCT13:
218:
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
OUTO,INPO,4,0,1,15,25,35,45

4316 016550
 4317 000002
 4318 016550 000600
 4319 016552
 4320 016552
 4321 000003
 4322 016552 062224
 4323 016554
 4324 000004
 4325 016554 023100
 4326 016556
 4327 000005
 4328 016556 063224
 4329 016560
 4330
 4331
 4332 016560
 4333 000006
 4334 016560 060360
 4335 016562
 4336 000007
 4337 016562 101422
 4338 016564
 4339 000010
 4340 016564 061224
 4341 016566
 4342 000011
 4343 016566 021105
 4344 016570
 4345 000012
 4346 016570 002401
 4347 016572
 4348 000013
 4349 016572 041223
 4350 016574
 4351 000014
 4352 016574 002404
 4353 016576
 4354 000015
 4355 016576 041227
 4356 016600
 4357 016600
 4358 000016
 4359 016600 000420
 4360 016602
 4361 000017
 4362 016602 104400
 4363 016604
 4364 000020
 4365 016604 060604
 4366 016606
 4367 000021
 4368 016606 100403
 4369 016610
 4370 016610
 4371 000022

```

15: MOVE # 200, BREG ;START WITH BIT 7.
    MICPC=MICPC+1
    .WORD .S.
25:
    MOVE BREG, OUT0 <4> ;SET THE BIT.
    MICPC=MICPC+1
    .WORD .S.
    MOVE INPO <4>, 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> 35 ;CHECK THE DATA...

SUBRC SPAD <0>, BREG, NOP
MICPC=MICPC+1
.WORD .S.
BZ 35
MICPC=MICPC+1
.WORD .S.
MOVE BREG, OUT1 <CSR4> ;GOOD DATA...
MICPC=MICPC+1
.WORD .S.
MOVE INPO <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
.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
.WORD .S.
CALL SCP1 ;IS LOOP DATA SET.
MOVE # <MICPC+3>, BREG
MICPC=MICPC+1
    
```

KMC11 MICRO PROCESSOR IBUS TESTS

4372	016610	000424	.WORD .S.	
4373	016612		SBR SCP1	
4374		000023	MICPC=MICPC+1	
4375	016612	104427	.WORD .S.	
4376	016614		MOVE SPAD <4>,BREG	
4377		000024	MICPC=MICPC+1	
4378	016614	060604	.WORD .S.	
4379	016616		SBR 25	;YES, DO IT.
4380		000025	MICPC=MICPC+1	
4381	016616	100403	.WORD .S.	
4382	016620		MOVE SPAD <4>,BREG	
4383		000026	MICPC=MICPC+1	
4384	016620	060604	.WORD .S.	
4385	016622		SHFBT	;NO, CONTINUE...
4386		000027	MICPC=MICPC+1	
4387	016622	061620	.WORD SBR!..SELB!..DBRSH	
4388	016624		B87 45	;IS IT DONE?...
4389		000030	MICPC=MICPC+1	
4390	016624	103432	.WORD .S.	
4391	016626		SBR 25	;NO, CONTINUE...
4392		000031	MICPC=MICPC+1	
4393	016626	100403	.WORD .S.	
4394	016630		45: \$FLOT	
4395	016630		OUTO INPO,4 0 0,115,125,135,145	
4396	016630		115: MOVE # 177,BREG ;START WITH BIT 7.	
4397		000032	MICPC=MICPC+1	
4398	016630	000577	.WORD .S.	
4399	016632		125: MOVE BREG,OUTO <4> ;SET THE BIT.	
4400	016632		MICPC=MICPC+1	
4401		000033	.WORD .S.	
4402	016632	062224	MOVE INPO <4>,SPAD <0>	;GET THE "FOUND" IN SCRATCH PAD.
4403	016634		MICPC=MICPC+1	
4404		000034	.WORD .S.	
4405	016634	023100	MOVE BREG,SPAD <4>	
4406	016636		MICPC=MICPC+1	
4407		000035	.WORD .S.	
4408	016636	063224	SIFEQ BREG,SPAD <0> 135	;CHECK THE DATA...
4409	016640			
4410				
4411				
4412	016640		SUB2C SPAD <0>,BREG,NOP	
4413		000036	MICPC=MICPC+1	
4414	016640	060360	.WORD .S.	
4415	016642		BZ 135	
4416		000037	MICPC=MICPC+1	
4417	016642	101452	.WORD .S.	
4418	016644		MOVE BREG,OUT1 <CSR4>	;GOOD DATA...
4419		000040	MICPC=MICPC+1	
4420	016644	061224	.WORD .S.	
4421	016646		MOVE INPO <4>,OUT1 <CSR5>;BAD DATA...	
4422		000041	MICPC=MICPC+1	
4423	016646	021105	.WORD .S.	
4424	016650		MOVE # 1, MEM	;TYPE OF ERROR...
4425		000042	MICPC=MICPC+1	
4426	016650	002401	.WORD .S.	
4427	016652		VE MEM,OUT1 <CSR3>	;

KMC11 MICRO PROCESSOR IBUS TESTS

4430	016653	000043	NICPC=NICPC+1	
4431	016654	002404	.WORD .S.	
4432	016654	000044	MOVE 8(4),MEM ;	
4433	016656	002404	NICPC=NICPC+1	
4434	016656	000045	.WORD .S.	
4435	016656	041227	MOVE MEM,OUT1 (CSR7) ;REG. ADDRESS.	
4436	016660	000046	NICPC=NICPC+1	
4437	016660	000450	.WORD .S.	
4438	016662	000047	SBR ERROR ;REPORT DATA ERROR.	
4439	016662	104400	NICPC=NICPC+1	
4440	016664	000050	.WORD .S.	
4441	016664	060604	MOVE SPAD (4),BREG ;RESTORE BREG...	
4442	016666	000051	NICPC=NICPC+1	
4443	016670	100433	.WORD .S.	
4444	016670	000052	CALL SCPI ;IS LOOP DATA SET.	135:
4445	016670	000454	MOVE 8 (NICPC+0),BREG	
4446	016672	000053	NICPC=NICPC+1	
4447	016672	104427	.WORD .S.	
4448	016674	000054	MOVE SPAD (4),BREG	
4449	016674	060604	NICPC=NICPC+1	
4450	016676	000055	.WORD .S.	
4451	016676	100433	SBR 125 ;YES, DO IT.	
4452	016700	000056	NICPC=NICPC+1	
4453	016700	060604	.WORD .S.	
4454	016702	000057	SHFBRT ;NO, CONTINUE...	
4455	016704	061620	NICPC=NICPC+1	
4456	016704	000060	.WORD .SBR!.SELB!.DBRSH ;	
4457	016704	103433	NICPC=NICPC+1	
4458	016706	000061	.WORD .S.	
4459	016706	000463	CALL SCPE	145:
4460	016706	000062	MOVE 8 (NICPC+3),BREG	
4461	016710	104454	NICPC=NICPC+1	
4462	016710	000063	.WORD .S.	
4463	016712	100400	SBR SCPE	
4464	016714		NICPC=NICPC+1	
4465	016714		.WORD .S.	
4466	016714		SBR 215	
4467			NICPC=NICPC+1	
4468			.WORD .S.	
4469			OUTO,INPO,5,0	\$FLT1
4470				\$XZ

```

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

; TEST 14
-----
TST14: MOV      #14,STSTNM          ; LOAD THE NO. OF THIS TEST
        MOV      @STST15,NEXT    ; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
        JSR      PC,LDRVMT      ;LOAD-VERIFY-WAIT.
MCT14  ERROR    22              ;TIME OUT ERROR...
        MOV      @STACK,SP      ;RESET STACK...
        JMP      @NEXT          ;GO TO NEXT TEST...

21S:    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
15:    MOVE     #200,BREG       ;START WITH BIT 7.
        MICPC=MICPC+1
        .WORD   .S.

25:    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.
SIFEQ  BREG,SPAD <0> 35       ;CHECK THE DATA...

SUB2C  SPAD <0>,BREG,NOP
        MICPC=MICPC+1
        .WORD   .S.
        BZ      35
        MICPC=MICPC+1
        .WORD   .S.
        MOVE     BREG,OUT1 <CSR4> ;GOOD DATA...
        MICPC=MICPC+1
        .WORD   .S.
        MOVE     INPO <5>,OUT1 <CSRS>;BAD DATA...
        MICPC=MICPC+1

```

```

4184
4185
4186
4187
4188
4189
4190 016714 $XZ
4191
4192 016714 $TSTN
4193
4194
4195 016714 012737 000014 001202 TST14:
4196 016722 012737 017120 001442 MOV
4197
4198
4199 016730 004737 035536 JSR
4200 016734 016750 MCT14
4201 016736 104022 ERROR
4202 016740 012706 001200 MOV
4203 016744 000177 162472 JMP
4204 016750 MCT14:
4205 016750 21S:
4206 016750 000000 MOVE
4207 016750 000400 MICPC=MICPC+1
4208 016752 000001 .WORD
4209 016752 063236 MOVE
4210 016752 063236 MICPC=MICPC+1
4211 016752 000002 .WORD
4212 016754 000600 SFLOT
4213 016754 000002 15:
4214 016754 000600 MOVE
4215 016754 062225 MICPC=MICPC+1
4216 016756 062225 .WORD
4217 016756 062225 MOVE
4218 016760 062225 MICPC=MICPC+1
4219 016760 023120 .WORD
4220 016762 000005 MOVE
4221 016762 063224 MICPC=MICPC+1
4222 016764 000006 .WORD
4223 016764 060360 SUB2C
4224 016766 000007 .WORD
4225 016766 101422 BZ
4226 016766 000007 MICPC=MICPC+1
4227 016766 101422 .WORD
4228 016770 000010 MOVE
4229 016770 061224 MICPC=MICPC+1
4230 016772 000011 .WORD
4231 016772 000011 MOVE
4232 016772 000011 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		S#BRT	;NO, CONTINUE...
4583		000027	MICPC=MICPC+1	
4584	017026	061620	.WORD SBR!.SELB!.DBRSH	
4585	017030		BB7 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	.WORD .S.	

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

12S:

4596	017036		MOVE	BREG,OUT0 <5>	;SET THE BIT.
4597	017036		NICPC=NICPC+1		
4598		000033	.WORD	.S.	
4599	017036	062225	MOVE	INPO <5>,SPAD <0>	;GET THE "FOUND" IN SCRATCH PAD.
4600	017040		NICPC=NICPC+1		
4601		000034	.WORD	.S.	
4602	017040	023120	MOVE	BREG,SPAD <4>	
4603	017042		NICPC=NICPC+1		
4604		000035	.WORD	.S.	
4605	017042	063224	SIFE0	BREG,SPAD <0>	13S ;CHECK THE DATA...
4606	017044				
4607					
4608					
4609	017044		SUBRC	SPAD <0>,BREG,NOP	
4610		000036	NICPC=NICPC+1		
4611	017044	060360	.WORD	.S.	
4612	017046		BZ	13S	
4613		000037	NICPC=NICPC+1		
4614	017046	101452	.WORD	.S.	
4615	017050		MOVE	BREG,OUT1 <CSR4>	;GOOD DATA...
4616		000040	NICPC=NICPC+1		
4617	017050	061224	.WORD	.S.	
4618	017052		MOVE	INPO <5>,OUT1 <CSRS>	;BAD DATA...
4619		000041	NICPC=NICPC+1		
4620	017052	021125	.WORD	.S.	
4621	017054		MOVE	8 1,MEM	;TYPE OF ERROR...
4622		000042	NICPC=NICPC+1		
4623	017054	002401	.WORD	.S.	
4624	017056		MOVE	MEM,OUT1 <CSR3>	;
4625		000043	NICPC=NICPC+1		
4626	017056	041223	.WORD	.S.	
4627	017060		MOVE	8 5,MEM	;
4628		000044	NICPC=NICPC+1		
4629	017060	002405	.WORD	.S.	
4630	017062		MOVE	MEM,OUT1 <CSR7>	;REG. ADDRESS.
4631		000045	NICPC=NICPC+1		
4632	017062	041227	.WORD	.S.	
4633	017064		CALL	ERROR	;REPORT DATA ERROR.
4634	017064		MOVE	8 <NICPC+3>,BREG	
4635		000046	NICPC=NICPC+1		
4636	017064	000450	.WORD	.S.	
4637	017066		SBR	ERROR	
4638		000047	NICPC=NICPC+1		
4639	017066	104400	.WORD	.S.	
4640	017070		MOVE	SPAD <4>,BREG	;RESTORE BREG...
4641		000050	NICPC=NICPC+1		
4642	017070	060604	.WORD	.S.	
4643	017072		SBR	12S	;LOOP ON ERROR...
4644		000051	NICPC=NICPC+1		
4645	017072	100433	.WORD	.S.	
4646	017074		CALL	SCP1	;IS LOOP DATA SET.
4647	017074		MOVE	8 <NICPC+3>,BREG	
4648		000052	NICPC=NICPC+1		
4649	017074	000454	.WORD	.S.	
4650	017076		SBR	SCP1	
4651		000053	NICPC=NICPC+1		

13S:

4652 017076 104427
4653 017100
4654 000054
4655 017100 060604
4656 017102
4657 000055
4658 017102 100433
4659 017104
4660 000056
4661 017104 060604
4662 017106
4663 000057
4664 017106 061620
4665 017110
4666 000060
4667 017110 103433
4668 017112
4669 017112
4670 017112
4671 000061
4672 017112 000463
4673 017114
4674 000062
4675 017114 104454
4676 017116
4677 000063
4678 017116 100400
4679 017120
4680 017120
4681
4682
4683
4684
4685
4686
4687 017120
4688
4689
4690 017120
4691
4692
4693 017120 012737 000015 001202
4694 017126 012737 017324 001442
4695
4696 017134 004737 035536
4697 017140 017154
4698 017142 104022
4699 017144 012706 001200
4700 017150 000177 162266
4701 017154
4702 017154
4703 017154
4704 000000
4705 017154 000400
4706 017156
4707 000001

```

.WORD      $
MOVE      SPAD (4),BREG
MICPC=MICPC+1
.WORD      $
SBR      12$
MICPC=MICPC+1
;YES, DO IT.
.WORD      $
MOVE      SPAD (4),BREG
MICPC=MICPC+1
.WORD      $
SHFBRT
MICPC=MICPC+1
;NO, CONTINUE...
.WORD      SBR!.SELB!.DBRSH
SBR      12$
MICPC=MICPC+1
.WORD      $
14$:
CALL      SCPE
MOVE      # <MICPC+3>,BREG
MICPC=MICPC+1
.WORD      $
SBR      SCPE
MICPC=MICPC+1
.WORD      $
SBR      21$
MICPC=MICPC+1
.SFLT1
OUTD,INPO,6,0
SXZ

;***** TEST 15 *****
;* MICRO PROCESSOR OUTD REGISTER WRITE/READ TEST.
;* FLOAT A 1 THROUGH REGISTER OUTD <6>
;* FLOAT A 0 THROUGH REGISTER OUTD <6>
SXZ
;*****

STSTN
; TEST 15
-----
TST15:  MOV      #15,STSTNM      ; LOAD THE NO. OF THIS TEST
        MOV      #ST16,NEXT    ; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
;LOAD-VERIFY-WAIT.
JSR      PC,LDVWNT
MCT15
ERROR    22
MOV      #STACK,SP
JMP      @NEXT
;TIME OUT ERROR...
;RESET STACK...
;GO TO NEXT TEST...

MCT15:
21$:
MOVE      # 0,BREG
MICPC=MICPC+1
;SET TO CLEAR SPAD 16
.WORD      $
MOVE      BREG,SPAD (16)
MICPC=MICPC+1
;FOR RETURN ADDRESS PURPOSES...

```

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
 0,1,15,25,35,45
 IS: MOVE # 200, BREG ; START WITH BIT 7.
 .WORD .S.
 2S: MOVE BREG, OUT0 <6> ; SET THE BIT.
 NICPC=NICPC+1
 .WORD .S.
 MOVE INFO <6>, 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> 3S ; CHECK THE DATA...
 SUBRC SPAD <0>, BREG, NOP
 NICPC=NICPC+1
 .WORD .S.
 BZ 3S
 NICPC=NICPC+1
 .WORD .S.
 MOVE BREG, OUT1 <CSR4> ; GOOD DATA...
 NICPC=NICPC+1
 .WORD .S.
 MOVE INFO <6>, OUT1 <CSRS>; BAD DATA...
 NICPC=NICPC+1
 .WORD .S.
 MOVE # 1, MEM ; TYPE OF ERROR...
 NICPC=NICPC+1
 .WORD .S.
 MOVE MEM, OUT1 <CSR3> ;
 NICPC=NICPC+1
 .WORD .S.
 MOVE # 6, MEM ;
 NICPC=NICPC+1
 .WORD .S.
 MOVE MEM, OUT1 <CSR7> ; REG. ADDRESS.
 NICPC=NICPC+1
 .WORD .S.
 CALL ERROR ; REPORT DATA ERROR.
 MOVE # <NICPC+3>, BREG
 NICPC=NICPC+1
 .WORD .S.
 SBR ERROR
 NICPC=NICPC+1
 .WORD .S.
 MOVE SPAD <4>, BREG ; RESTORE BREG...
 NICPC=NICPC+1
 .WORD .S.
 SBR 2S ; LOOP ON ERROR...
 NICPC=NICPC+1
 .WORD .S.
 3S: CALL SCP1 ; IS LOOP DATA SET.

4764	017220		MOVE # <MICPC+3>,BREG	
4765		000022	MICPC=MICPC+1	
4766	017220	000424	.WORD .S.	
4767	017222		SBR SCP1	
4768		000023	MICPC=MICPC+1	
4769	017222	104427	.WORD .S.	
4770	017224		MOVE SPAD <4>,BREG	
4771		000024	MICPC=MICPC+1	
4772	017224	060604	.WORD .S.	
4773	017226		SBR 25	;YES, DO IT.
4774		000025	MICPC=MICPC+1	
4775	017226	100403	.WORD .S.	
4776	017230		MOVE SPAD <4>,BREG	
4777		000026	MICPC=MICPC+1	
4778	017230	060604	.WORD .S.	
4779	017232		SHFBRT	;NO, CONTINUE...
4780		000027	MICPC=MICPC+1	
4781	017232	061620	.WORD SBR!.SELB!.DBRSH	
4782	017234		BZ 45	;IS IT DONE?...
4783		000030	MICPC=MICPC+1	
4784	017234	103432	.WORD .S.	
4785	017236		SBR 25	;NO, CONTINUE...
4786		000031	MICPC=MICPC+1	
4787	017236	100403	.WORD .S.	
4788	017240			
4789	017240		45: SFL0T OUT0,INP0,6,0,0,115,125,135,145	
4790	017240		115: MOVE # 177,BREG ;START WITH BIT 7.	
4791		000032	MICPC=MICPC+1	
4792	017240	000577	.WORD .S.	
4793	017242			
4794	017242		125: MOVE BREG,OUT0 <6> ;SET THE BIT.	
4795		000033	MICPC=MICPC+1	
4796	017242	062226	.WORD .S.	
4797	017244		MOVE INP0 <6>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.	
4798		000034	MICPC=MICPC+1	
4799	017244	023140	.WORD .S.	
4800	017246		MOVE BREG,SPAD <4>	
4801		000035	MICPC=MICPC+1	
4802	017246	063224	.WORD .S.	
4803	017250		SIFEQ BREG,SPAD <0> 135 ;CHECK THE DATA...	
4804				
4805				
4806	017250		SUB2C SPAD <0>,BREG,NOP	
4807		000036	MICPC=MICPC+1	
4808	017250	060360	.WORD .S.	
4809	017252		BZ 135	
4810		000037	MICPC=MICPC+1	
4811	017252	101452	.WORD .S.	
4812	017254		MOVE BREG,OUT1 <CSR4> ;GOOD DATA...	
4813		000040	MICPC=MICPC+1	
4814	017254	061224	.WORD .S.	
4815	017256		MOVE INP0 <6>,OUT1 <CSR5>;BAD DATA...	
4816		000041	MICPC=MICPC+1	
4817	017256	021145	.WORD .S.	
4818	017260		MOVE # 1,MEM ;TYPE OF ERROR...	
4819		000042	MICPC=MICPC+1	

4820	017260	002401	.WORD .S.	
4821	017262		MOVE MEM,OUT1 <CSR3>	;
4822		000043	MICPC=MICPC+1	
4823	017262	041223	.WORD .S.	
4824	017264		MOVE 8 6 MEM	;
4825		000044	MICPC=MICPC+1	
4826	017264	002406	.WORD .S.	
4827	017266		MOVE MEM,OUT1 <CSR7>	;REG. ADDRESS.
4828		000045	MICPC=MICPC+1	
4829	017266	041227	.WORD .S.	
4830	017270		CALL ERROR	;REPORT DATA ERROR.
4831	017270		MOVE 8 <MICPC+3>,BREG	
4832		000046	MICPC=MICPC+1	
4833	017270	000450	.WORD .S.	
4834	017272		SBR ERROR	
4835		000047	MICPC=MICPC+1	
4836	017272	104430	.WORD .S.	
4837	017274		MOVE SPAD <4>,BREG	;RESTORE BREG...
4838		000050	MICPC=MICPC+1	
4839	017274	060604	.WORD .S.	
4840	017276		SBR 125	;LOOP ON ERROR...
4841		000051	MICPC=MICPC+1	
4842	017276	100433	.WORD .S.	
4843	017300		CALL SCP1	;IS LOOP DATA SET.
4844	017300		MOVE 8 <MICPC+3>,BREG	
4845		000052	MICPC=MICPC+1	
4846	017300	000454	.WORD .S.	
4847	017302		SBR SCP1	
4848		000053	MICPC=MICPC+1	
4849	017302	104427	.WORD .S.	
4850	017304		MOVE SPAD <4>,BREG	
4851		000054	MICPC=MICPC+1	
4852	017304	060604	.WORD .S.	
4853	017306		SBR 125	;YES, DO IT.
4854		000055	MICPC=MICPC+1	
4855	017306	100433	.WORD .S.	
4856	017310		MOVE SPAD <4>,BREG	
4857		000056	MICPC=MICPC+1	
4858	017310	060604	.WORD .S.	
4859	017312		SFBRT	;NO, CONTINUE...
4860		000057	MICPC=MICPC+1	
4861	017312	061620	.WORD SBR!.SELB!.DBRSH	
4862	017314		BB7 125	;
4863		000060	MICPC=MICPC+1	
4864	017314	103433	.WORD .S.	
4865	017316		CALL SCPE	
4866	017316		MOVE 8 <MICPC+3>,BREG	
4867	017316		MICPC=MICPC+1	
4868		000061	.WORD .S.	
4869	017316	000463	SBR SCPE	
4870	017320		MICPC=MICPC+1	
4871		000062	.WORD .S.	
4872	017320	104454	SBR 215	
4873	017322		MICPC=MICPC+1	
4874		000063	.WORD .S.	
4875	017322	100400		

135:

145:

KMC11 MICRO PROCESSOR IBUS TESTS

```

4876 017324 $FLT1 OUT0,INPO,7,0
4877 017324 $XZ
4878
4879
4880 ;***** TEST 16 *****
4881 ;* MICRO PROCESSOR OUT0 REGISTER WRITE/READ TEST.
4882 ;* FLOAT A 1 THROUGH REGISTER OUT0 <7>
4883 ;* FLOAT A 0 THROUGH REGISTER OUT0 <7>
4884 017324 $XZ ;:*****
4885
4886
4887 017324 $TSTN ; TEST 16
4888 ;-----
4889
4890 017324 012737 000016 001202 TST16: MOV #16,$TSTNM ; LOAD THE NO. OF THIS TEST
4891 017332 012737 017530 001442 MOV #TST17,NEXT ; POINT TO THE START OF NEXT TEST.
4892 ;R1 CONTAINS BASE KMC11 ADDRESS
4893 ;LOAD-VERIFY-WAIT.
4894 017340 004737 035536 JSR PC,LDRWT
4895 017344 017360 NCT16
4896 017346 104022 ERROR 22 ;TIME OUT ERROR...
4897 017350 012705 001200 MOV #STACK,SP ;RESET STACK
4898 017354 000177 162062 JMP @NEXT ;GO TO NEXT TEST...
4899
4900 017360 NCT16:
4901 017360 000000 MOVE #0,BREG ;SET TO CLEAR SPAD 16
4902 017360 000400 MICPC=MICPC+1
4903 017362 .WORD $
4904 017362 000001 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES...
4905 017362 063236 MICPC=MICPC+1
4906 017364 $FLOT OUT0,INPO,7,0,1,15,25,35,45
4907 017364 1S: MOVE #200,BREG ;START WITH BIT 7.
4908 017364 000002 MICPC=MICPC+1
4909 017364 000600 .WORD $.
4910 017366 2S:
4911 017366 000003 MOVE BREG,OUT0 <7> ;SET THE BIT.
4912 017366 062227 MICPC=MICPC+1
4913 017366 062227 .WORD $
4914 017370 000004 MOVE INPO <7>,SPAD <0> ;GET THE "FOUND" IN SCRATCH PAD.
4915 017370 023160 MICPC=MICPC+1
4916 017370 000005 .WORD $
4917 017372 063224 MOVE BREG,SPAD <4>
4918 017372 063224 MICPC=MICPC+1
4919 017374 063224 .WORD $
4920 017374 3S: SIFEQ BREG,SPAD <0> 3S ;CHECK THE DATA...
4921
4922
4923 017374 SUB2C SPAD <0>,BREG,NOP
4924 017374 000006 MICPC=MICPC+1
4925 017374 060360 .WORD $
4926 017376 BZ 3S
4927 017376 000007 MICPC=MICPC+1
4928 017376 101422 .WORD $
4929 017400 MOVE BREG,OUT1 <CSR4> ;GOOD DATA...
4930 017400 000010 MICPC=MICPC+1
4931 017400 061224 .WORD $.

```

KMC11 MICRO PROCESSOR IBUS TESTS

```

4932 017402          MOVE      INPO (>),OUT1 (CSRS);BAD DATA...
4933          MICPC=MICPC+1
4934 017402 000011   .WORD     .S.
4935 017404          MOVE      # 1, MEM ;TYPE OF ERROR...
4936          MICPC=MICPC+1
4937 017404 000012   .WORD     .S.
4938 017406          MOVE      MEM,OUT1 (CSR3) ;
4939          MICPC=MICPC+1
4940 017406 041223   .WORD     .S.
4941 017410          MOVE      # 7, MEM ;
4942          MICPC=MICPC+1
4943 017410 000013   .WORD     .S.
4944 017412          MOVE      MEM,OUT1 (CSR7) ;REG. ADDRESS.
4945          MICPC=MICPC+1
4946 017412 041227   .WORD     .S.
4947 017414          CALL      EROR ;REPORT DATA ERROR.
4948 017414          MOVE      # (MICPC+3),BREG
4949          MICPC=MICPC+1
4950 017414 000016   .WORD     .S.
4951 017416          SBR      EROR
4952          MICPC=MICPC+1
4953 017416 104400   .WORD     .S.
4954 017420          MOVE      SPAD (<4>),BREG ;RESTORE BREG...
4955          MICPC=MICPC+1
4956 017420 060604   .WORD     .S.
4957 017422          SBR      25 ;LOOP ON ERROR...
4958          MICPC=MICPC+1
4959 017422 000021   .WORD     .S.
4960 017424          CALL      SCP1 ;IS LOOP DATA SET.
4961 017424          MOVE      # (MICPC+3),BREG
4962          MICPC=MICPC+1
4963 017424 000022   .WORD     .S.
4964 017426          SBR      SCP1
4965          MICPC=MICPC+1
4966 017426 104427   .WORD     .S.
4967 017430          MOVE      SPAD (<4>),BREG
4968          MICPC=MICPC+1
4969 017430 060604   .WORD     .S.
4970 017432          SBR      25 ;YES, DO IT.
4971          MICPC=MICPC+1
4972 017432 000025   .WORD     .S.
4973 017434          MOVE      SPAD (<4>),BREG
4974          MICPC=MICPC+1
4975 017434 060604   .WORD     .S.
4976 017436          SHFBRT ;NO, CONTINUE...
4977          MICPC=MICPC+1
4978 017436 061620   .WORD     .SBR!.SELB!.DBRSH
4979 017440          BBT      45 ;IS IT DONE?...
4980          MICPC=MICPC+1
4981 017440 103432   .WORD     .S.
4982 017442          SBR      25 ;NO, CONTINUE...
4983          MICPC=MICPC+1
4984 017442 100403   .WORD     .S.
4985 017444          SFL0T   OUT0,INPO,7,0,0,11$,12$,13$,14$
4986 017444          11$: MOVE # 177,BREG ;START WITH BIT 7.
4987 017444

```

KMC11 MICRO PROCESSOR IBUS TESTS

4988		000032	MICPC=MICPC+1	
4989	017444	000577	.WORD .S.	
4990	017446			
4991	017446		12S: MOVE BREG,OUT0 (>) ;SET THE BIT.	
4992		000033	MICPC=MICPC+1	
4993	017446	062227	.WORD .S.	
4994	017450		MOVE INPO (>),SPAD (<0) ;GET THE "FOUND" IN SCRATCH PAD.	
4995		000034	MICPC=MICPC+1	
4996	017450	023160	.WORD .S.	
4997	017452		MOVE BREG,SPAD (<4)	
4998		000035	MICPC=MICPC+1	
4999	017452	063224	.WORD .S.	
5000	017454		SIFEQ BREG,SPAD (<0) 13S ;CHECK THE DATA...	
5001				
5002				
5003	017454		SUBRC SPAD (<0),BREG,NOP	
5004		000036	MICPC=MICPC+1	
5005	017454	060360	.WORD .S.	
5006	017456		BZ 13S	
5007		000037	MICPC=MICPC+1	
5008	017456	101452	.WORD .S.	
5009	017460		MOVE BREG,OUT1 (CSR4) ;GOOD DATA...	
5010		000040	MICPC=MICPC+1	
5011	017460	061224	.WORD .S.	
5012	017462		MOVE INPO (>),OUT1 (CSRS);BAD DATA...	
5013		000041	MICPC=MICPC+1	
5014	017464	021165	.WORD .S.	
5015		000042	MOVE 8,1,MEM ;TYPE OF ERROR...	
5016		002401	MICPC=MICPC+1	
5017	017466		.WORD .S.	
5018		000043	MOVE MEM,OUT1 (CSR3) ;	
5019		000043	MICPC=MICPC+1	
5020	017466	041223	.WORD .S.	
5021	017470		MOVE 8,7,MEM ;	
5022		000044	MICPC=MICPC+1	
5023	017470	002407	.WORD .S.	
5024	017472		MOVE MEM,OUT1 (CSR7) ;REG. ADDRESS.	
5025		000045	MICPC=MICPC+1	
5026	017472	041227	.WORD .S.	
5027	017474		CALL ERROR ;REPORT DATA ERROR.	
5028	017474		MOVE 8 (MICPC+3),BREG	
5029		000046	MICPC=MICPC+1	
5030	017474	000450	.WORD .S.	
5031	017476		CALL ERROR	
5032		000047	MICPC=MICPC+1	
5033	017476	104400	.WORD .S.	
5034	017500		MOVE SPAD (<4),BREG ;RESTORE BREG...	
5035		000050	MICPC=MICPC+1	
5036	017500	060604	.WORD .S.	
5037	017502		SRR 12S ;LOOP ON ERROR...	
5038		000051	MICPC=MICPC+1	
5039	017502	100433	.WORD .S.	
5040	017504		13S: CALL SCP1 ;IS LOOP DATA SET.	
5041	017504		MOVE 8 (MICPC+3),BREG	
5042		000052	MICPC=MICPC+1	
5043	017504	000454	.WORD .S.	

MACY11 27(1006) 13-MAY-77 14:07 PAGE 97
JZCA.P11 13-MAY-77 13:58

KMC11 MICRO PROCESSOR IBUS TESTS

5044 017506 000053
5045 017506 104427
5046 017510 000054
5047 017510 060604
5048 017512 000055
5049 017512 100433
5050 017514 000056
5051 017514 060604
5052 017516 000057
5053 017516 061620
5054 017520 000060
5055 017520 103433
5056 017522 000061
5057 017522 000463
5058 017524 000062
5059 017524 104454
5060 017526 000063
5061 017526 100400
5062 017530
5063 017530
5064 017530
5065 017530
5066 017530
5067 017530
5068 017530 012737 000017 001202
5069 017536 012737 017720 001442
5070 017544 004737 035536
5071 017550 017564
5072 017552 104022
5073 017554 012706 001200
5074 017560 000177 161656
5075 017564
5076 017564
5077 017564
5078 017564
5079 017564
5080 017564
5081 017564
5082 017564
5083 017564
5084 017564
5085 017564
5086 017564
5087 017564
5088 017564
5089 017564
5090 017564
5091 017564
5092 017564
5093 017564
5094 017564
5095 017564
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      12$
;YES, DO IT.
MICPC=MICPC+1
.WORD   .S.
MOVE    SPAD (4),BREG
MICPC=MICPC+1
.WORD   .S.
SHIFT   .S.
;NO, CONTINUE...
MICPC=MICPC+1
.WORD   .S.
SBR      SBR!.SELB!.DBRSH
BB7     12$
MICPC=MICPC+1
.WORD   .S.

14$:
CALL    SCPE
MOVE    # (MICPC+3),BREG
MICPC=MICPC+1
.WORD   .S.
SBR      SCPE
MICPC=MICPC+1
.WORD   .S.
SBR      21$
MICPC=MICPC+1
.WORD   .S.

SPROC4
SXZ

;***** TEST 17 *****
; MICRO PROCESSOR B REGISTER TEST
; FLOAT A I THROUGH THE BREG.
; FLOAT A O THROUGH THE BREG.
;
SXZ
;*****

STSTN
; TEST 17
-----
TST17:  MOV    #17,STSTNM
; LOAD THE NO. OF THIS TEST
MOV    #STST20,NEXT
; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
;LOAD-VERIFY-WAIT.

JSR    PC,LDRWMT
MCT17  ERROR  22
;TIME OUT ERROR...
MOV    #STACK,SP
;RESET STACK...
JMP    @NEXT
;GO TO NEXT TEST...

21$:
MOVE    # 0,BREG
;SET TO CLEAR SPAD 16.
MICPC=MICPC+1

```

DZXCA MACY11 27(1006) 13-MAY-77 14:07 PAGE 98
DZXCA.P11 13-MAY-77 13:58

KMC11 MICRO PROCESSOR IBUS TESTS

```

5100 017564 000400      .WORD      $
5101 017566      MOVE      BREG,SPAD (16) ;FOR RETURN ADDRESS PURPOSE...
5102      MICPC=MICPC+1
5103 017566 063236      .WORD      $
5104 017570      SBRFLT   i 15,25,35,45
5105 017570      15:      MOVE      # 001, MEM ;START WITH BIT0.
5106      MICPC=MICPC+1
5107 017570 002401      .WORD      $
5108 017572      MOVE      MEM,SPAD (2) ;SET THE DATA IN SPAD0.
5109      MICPC=MICPC+1
5110 017572 000003      .WORD      $
5111 017574      25:      MOVE      MEM,BREG ;SET THE BIT IN BREG.
5112      MICPC=MICPC+1
5113 017574 040620      .WORD      $
5114 017576      MOVE      BREG,SPAD (1) ;PUT IT IN SPAD (1).
5115      MICPC=MICPC+1
5116 017576 000005      .WORD      $
5117 017600      SIFEQ   MEM,SPAD (1) 35 ;COMPARE DATA...
5118
5119
5120 017600      SUB2C   SPAD (1),MEM,NOP
5121      MICPC=MICPC+1
5122 017600 040361      .WORD      $
5123 017602      BZ      $
5124      MICPC=MICPC+1
5125 017602 000007      .WORD      $
5126 017604      MOVE      MEM,OUT1 (CSR4) ;REPORT DATA ERROR.
5127      MICPC=MICPC+1
5128 017604 000010      .WORD      $
5129 017606      MOVE      BREG,OUT1 (CSR5) ; - - -
5130      MICPC=MICPC+1
5131 017606 061225      .WORD      $
5132 017610      MOVE      # 3,OUT1 (CSR3) ;ERROR TYPE...
5133      MICPC=MICPC+1
5134 017610 000012      .WORD      $
5135 017612      CALL     ERROR1 ;DATA ERROR...
5136 017612      MOVE      # <MICPC+3>,BREG
5137      MICPC=MICPC+1
5138 017612 000013      .WORD      $
5139 017614      SER     ERROR1
5140      MICPC=MICPC+1
5141 017614 000014      .WORD      $
5142 017616      SBR     ;LOOP ON ERROR...
5143      MICPC=MICPC+1
5144 017616 000015      .WORD      $
5145 017620      33:      CALL     SCP11 ;LOOP THE DATA...
5146 017620      MOVE      # <MICPC+3>,BREG
5147      MICPC=MICPC+1
5148 017620 000016      .WORD      $
5149 017622      SBR     SCP11
5150      MICPC=MICPC+1
5151 017622 000017      .WORD      $
5152 017624      SBR     25 ;YES, LOOP...
5153      MICPC=MICPC+1
5154 017624 100404      .WORD      $
5155 017626      MOVE      MEM,BREG

```

KMC11 MICRO PROCESSOR IBUS TESTS

```

5151 017626 000021      NICPC=NICPC+1
5152 017626 040620      .WORD .S.
5153 017630 000022      SROL SPAD (2) ;SET THE NEXT BIT.
5154 017630 063142      NICPC=NICPC+1
5155 017632 000023      .WORD .S. DSP
5156 017632 063102      SROL SPAD (2)
5157 017634 000024      NICPC=NICPC+1
5158 017634 062602      .WORD .S. DSP
5159 017634 062602      MOVE SPAD (2),MEM ;
5160 017636 000025      NICPC=NICPC+1
5161 017636 103427      .WORD .S.
5162 017640 000026      SRR ;DONE...
5163 017640 100404      NICPC=NICPC+1
5164 017642 000027      .WORD .S.
5165 017642 002776      SRRFLT 0,118,128,138,148 ;START WITH BIT0.
5166 017644 000030      MOVE #376,MEM
5167 017644 043222      NICPC=NICPC+1
5168 017644 000031      .WORD .S.
5169 017646 000032      MOVE MEM,SPAD (2) ;SET THE DATA IN SPAD0.
5170 017646 040620      NICPC=NICPC+1
5171 017650 000033      .WORD .S.
5172 017652 000034      MOVE MEM,BREG ;SET THE BIT IN BREG.
5173 017654 000035      NICPC=NICPC+1
5174 017654 041224      .WORD .S.
5175 017656 000036      MOVE BREG,SPAD (1) ;PUT IT IN SPAD (1).
5176 017656 063221      NICPC=NICPC+1
5177 017658 000037      .WORD .S.
5178 017658 061225      SIFEQ MEM,SPAD (1) 138 ;COMPARE DATA...
5179 017658 000038      SUB2C SPAD (1),MEM,NOP
5180 017658 040361      NICPC=NICPC+1
5181 017658 000039      .WORD .S.
5182 017658 101443      BZ 138
5183 017658 000040      NICPC=NICPC+1
5184 017658 041224      .WORD .S.
5185 017658 000041      MOVE MEM,OUT1 (CSR4) ;REPORT DATA ERROR.
5186 017658 041224      NICPC=NICPC+1
5187 017658 000042      .WORD .S.
5188 017658 000043      MOVE BREG,OUT1 (CSR5) ; " " " "
5189 017658 061225      NICPC=NICPC+1
5190 017658 000044      .WORD .S.
5191 017658 001003      MOVE #3,OUT1 (CSR3) ;ERROR TYPE...
5192 017658 000045      NICPC=NICPC+1
5193 017658 000046      .WORD .S.
5194 017658 000047      CALL ERROR1 ;DATA ERROR...
5195 017658 000048      MOVE # (NICPC+3),BREG
5196 017658 000049      NICPC=NICPC+1
5197 017658 000050      .WORD .S.
5198 017658 000051      SRR ERROR1
5199 017658 000052      NICPC=NICPC+1
5200 017658 104401      .WORD .S.
    
```

KMC11 MICRO PROCESSOR IBUS TESTS

```

017670 000042          SER      125          ;LOOP ON ERROR...
017670 100431          MICPC=MICPC+1
017672          .WORD      .S.
017672          135:      CALL      SCP11          ;LOOP THE DATA...
017672          MOVE      # <MICPC+3>,BREG
017672          000043          MICPC=MICPC+1
017672          000044          .WORD      .S.
017674          000044          SER      SCP11
017674          104430          MICPC=MICPC+1
017676          000045          .WORD      .S.          ;YES, LOOP...
017676          100431          SER      125
017700          000045          MICPC=MICPC+1
017700          040620          .WORD      .S.
017702          000046          MOVE      MEM,BREG
017702          040620          MICPC=MICPC+1
017702          063142          .WORD      .S.          ;SET THE NEXT BIT.
017704          000050          .WORD      .S.          .S.!.DSP
017704          063102          .WORD      .S.          .S.!.DSP
017706          000051          MOVE      SPAD <2>,MEM          ;
017706          062602          MICPC=MICPC+1
017710          000052          .WORD      .S.
017710          103431          887      125          ;CONTINUE, IF NOT DONE...
017712          000053          MICPC=MICPC+1
017712          000455          .WORD      .S.
017714          000054          CALL      SCPE
017714          104454          MOVE      # <MICPC+3>,BREG
017716          000055          MICPC=MICPC+1
017716          100400          .WORD      .S.
017720          $SPTS1 4
017720          $XZ

;***** TEST 20 *****
;* SCRATCH PAD TEST FOR SP4
;* FLOAT A 1 THROUGH SCRATCH PAD 4
;* FLOAT A 0 THROUGH SCRATCH PAD 4
017720          $XZ
;*****

017720          $TSTN
; TEST 20
-----
017720 012737 000020 001202 TST20: MOV      #20,$TSTNM          ; LOAD THE NO. OF THIS TEST
017726 012737 020134 001442      MOV      #TST21,NEXT          ; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS

```

KMC11 SCRATCH PAD TESTS

5254	017724	004737	035536	JSR	PC,LDVRWT	;LOAD-VERIFY-WAIT.
5255	017740	017754		MCT20		
5270	017742	104022		ERROR	22	;TIME OUT ERROR...
5271	017744	012708	001200	MOV	#STACK,SP	;RESET STACK...
5272	017750	000177	161466	JMP	@NEXT	;GO TO NEXT TEST...
5273	017754					
5274	017754					
5275	017754					
5276		000000		MOVE	# 0,BREG	;SET TO CLEAR SPAD 16...
5277	017754	000400		NICPC=NICPC+1		
5278	017756			.WORD	.S.	
5279				MOVE	BREG,SPAD <16>	;FOR RETURN ADDRESS PURPOSES.
5280		000001		NICPC=NICPC+1		
5281	017756	063236		.WORD	.S.	
5282	017760			SSPFLT	1,18,28,38,48,4	
5283	017760			IS:	MOVE # 200,BREG	;START WITH BIT 7.
5284		000002		NICPC=NICPC+1		
5285	017760	000600		.WORD	.S.	
5286	017762			25:	MOVE BREG,SPAD <4>	;LOAD THE SCRATCH PAD...
5287		000003		NICPC=NICPC+1		
5288	017762	063224		.WORD	.S.	
5289	017764			MOVE	SPAD <4>,MEM	;GET THE "FOUND"...
5290		000004		NICPC=NICPC+1		
5291	017764	062604		.WORD	.S.	
5292	017766			MOVE	MEM,SPAD <0>	;
5293		000005		NICPC=NICPC+1		
5294	017766	043220		.WORD	.S.	
5295	017770			MOVE	BREG,MEM	;SAVE THE CONTENTS OF BREG...
5296		000006		NICPC=NICPC+1		
5297	017770	062620		.WORD	.S.	
5298	017772			SIFEQ	BREG,SPAD <0> 3\$;IF GOOD.. CONTINUE...
5299						
5300	017772			SUB2C	SPAD <0>,BREG,NOP	
5301		000007		NICPC=NICPC+1		
5302	017772	060360		.WORD	.S.	
5303	017774			BZ	3\$	
5304		000010		NICPC=NICPC+1		
5305	017774	101424		.WORD	.S.	
5306	017776			MOVE	BREG,OUT1 <CSR4>	;ELSE, REPORT ERROR...
5307		000011		NICPC=NICPC+1		
5308	017776	061224		.WORD	.S.	
5309	020000			MOVE	SPAD <0>,BREG	;
5310		000012		NICPC=NICPC+1		
5311	020000	060600		.WORD	.S.	
5312	020002			MOVE	BREG,OUT1 <CSR5>	;BAD DATA...
5313		000013		NICPC=NICPC+1		
5314	020002	061225		.WORD	.S.	
5315	020004			MOVE	# 4,BREG	;TYPE OF ERROR...
5316		000014		NICPC=NICPC+1		
5317	020004	000404		.WORD	.S.	
5318	020006			MOVE	BREG,OUT1 <CSR3>	;
5319		000015		NICPC=NICPC+1		
5320	020006	061223		.WORD	.S.	
5321	020010			MOVE	# 4,BREG	;LOAD ADDRESS.
5322		000016		NICPC=NICPC+1		
5323	020010	000404		.WORD	.S.	

5370	020012	000017	MOVE BREG,OUT1 <CSR7>	;
5371	020012	061227	MICPC=MICPC+1	
5372	020014		.WORD .S.	
5373	020014		CALL ERROR	;DATA ERROR!!
5374	020014		MOVE # <MICPC+3>,BREG	
5375	020014	000020	MICPC=MICPC+1	
5376	020014	000422	.WORD .S.	
5377	020016		SBR ERROR	
5378	020016	000021	MICPC=MICPC+1	
5379	020016	104400	.WORD .S.	
5380	020020		MOVE MEM,BREG	;RESTORE BREG...
5381	020020	000022	MICPC=MICPC+1	
5382	020020	040620	.WORD .S.	
5383	020022		SBR 25	;LOOP ON ERROR...
5384	020022	000023	MICPC=MICPC+1	
5385	020022	100403	.WORD .S.	
5386	020024		CALL SCP1	;IS LOOP DATA SET???
5387	020024		MOVE # <MICPC+3>,BREG	
5388	020024	000024	MICPC=MICPC+1	
5389	020024	000426	.WORD .S.	
5390	020026		SBR SCP1	
5391	020026	000025	MICPC=MICPC+1	
5392	020026	104427	.WORD .S.	
5393	020030		MOVE MEM,BREG	;RESTORE BREG...
5394	020030	000026	MICPC=MICPC+1	
5395	020030	040620	.WORD .S.	
5396	020032		SBR 25	;LOOP ON DATA.
5397	020032	000027	MICPC=MICPC+1	
5398	020032	100403	.WORD .S.	
5399	020034		MOVE MEM,BREG	;RESTORE BREG...
5400	020034	000030	MICPC=MICPC+1	
5401	020034	040620	.WORD .S.	
5402	020036		SHFBT	;SET THE NEXT BIT...
5403	020036	000031	MICPC=MICPC+1	
5404	020036	061620	.WORD .SBR!.SELB!.DBRSH	
5405	020040		B87 45	;BRANCH IF DONE...
5406	020040	000032	MICPC=MICPC+1	
5407	020040	103434	.WORD .S.	
5408	020042		SBR 25	;CONTINUE...
5409	020042	000033	MICPC=MICPC+1	
5410	020042	100403	.WORD .S.	
5411	020044		45:	
5412	020044		SSPFLT 0,115,125,135,145,4	
5413	020044		115: MOVE # 177,BREG ;START WITH BIT 7.	
5414	020044	000034	MICPC=MICPC+1	
5415	020044	000577	.WORD .S.	
5416	020046		125: MOVE BREG,SPAD <4>	;LOAD THE SCRATCH PAD...
5417	020046	000035	MICPC=MICPC+1	
5418	020046	063224	.WORD .S.	
5419	020050		MOVE SPAD <4>,MEM	;GET THE "FOUND"...
5420	020050	000036	MICPC=MICPC+1	
5421	020050	062604	.WORD .S.	
5422	020052		MOVE MEM,SPAD <0>	;
5423	020052	000037	MICPC=MICPC+1	
5424	020052	043220	.WORD .S.	
5425	020054		MOVE BREG,MEM	;SAVE THE CONTENTS OF BREG...

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100
000101
000102
000103
000104
000105
000106
000107
000108
000109
000110
000111
000112
000113
000114
000115
000116
000117
000118
000119
000120
000121
000122
000123
000124
000125
000126
000127
000128
000129
000130
000131
000132
000133
000134
000135
000136
000137
000138
000139
000140
000141
000142
000143
000144
000145
000146
000147
000148
000149
000150
000151
000152
000153
000154
000155
000156
000157
000158
000159
000160
000161
000162
000163
000164
000165
000166
000167
000168
000169
000170
000171
000172
000173
000174
000175
000176
000177
000178
000179
000180
000181
000182
000183
000184
000185
000186
000187
000188
000189
000190
000191
000192
000193
000194
000195
000196
000197
000198
000199
000200

020054 062620
020056 060360
020058 101456
020062 061224
020064 060600
020066 061225
020070 000404
020072 061223
020074 000404
020076 061227
020100 000454
020102 104400
020104 040620
020106 100435
020110 000460
020112 104427
020114 040620
020116

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

SUB2C SPAD <0>,BREG,NOP
MICPC=MICPC+1
WORD .S
BZ 135
MICPC=MICPC+1
WORD .S
MOVE BREG,OUT1 <CSR4> ;ELSE, REPORT ERROR...
MICPC=MICPC+1
WORD .S
MOVE SPAD.<0>,BREG ;
MICPC=MICPC+1
WORD .S
MOVE BREG,OUT1 <CSRS> ;BAD DATA...
MICPC=MICPC+1
WORD .S
MOVE # 4,BREG ;TYPE OF ERROR...
MICPC=MICPC+1
WORD .S
MOVE BREG,OUT1 <CSR3> ;
MICPC=MICPC+1
WORD .S
MOVE # 4,BREG ;LOAD ADDRESS.
MICPC=MICPC+1
WORD .S
MOVE BREG,OUT1 <CSR7> ;
MICPC=MICPC+1
WORD .S
CALL EROR ;DATA ERROR!!
MOVE # <MICPC+3>,BREG
MICPC=MICPC+1
WORD .S
SBR EROR
MICPC=MICPC+1
WORD .S
MOVE MEM,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 # <MICPC+3>,BREG
MICPC=MICPC+1
WORD .S
SBR SCP1
MICPC=MICPC+1
WORD .S
MOVE MEM,BREG ;RESTORE BREG...
MICPC=MICPC+1
WORD .S
SBR 125 ;LOOP ON DATA.
```

135:

KMC11 SCRATCH PAD TESTS

```

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 .SRR!..SELB!..DBRSH
5444 020124          BBT 128          ;BRANCH IF NOT DONE...
5445          000064          MICPC=MICPC+1
5446 020124 103435          .WORD .S.
5447          145:
5448 020126          CALL    SCPE
5449 020128          MOVE    # <MICPC+3>,BREG
5450          000065          MICPC=MICPC+1
5451 020126 000467          .WORD .S.
5452 020130          SRR    SCPE
5453          000066          MICPC=MICPC+1
5454 020130 104454          .WORD .S.
5455 020132          SRR    21$
5456          000067          MICPC=MICPC+1
5457 020132 100400          .WORD .S.
5458          $SPTS1 5
5459 020134          $XZ
5460
5461          ;***** TEST 21 *****
5462          ;* SCRATCH PAD TEST FOR SPS
5463          ;* FLOAT A 1 THROUGH SCRATCH PAD 5
5464          ;* FLOAT A 0 THROUGH SCRATCH PAD 5
5465          $XZ
5466          ;*****
5467
5468          $STSTN
5469 020134          ; TEST 21
5470          ;-----
5471          TST21: MOV    #21,$STSTN          ; LOAD THE NO. OF THIS TEST
5472 020134 012737 000021 001202          MOV    #TST22,NEXT          ; POINT TO THE START OF NEXT TEST.
5473 020142 012737 020350 001442          ;R1 CONTAINS BASE KMC11 ADDRESS
5474          JSR    PC,LDRANT          ;LOAD-VERIFY-WAIT.
5475 020150 004737 035536          MCT21
5476 020154 020170          ERROR    22          ;TIME OUT ERROR...
5477 020156 104022          MOV    #STACK,SP          ;RESET STACK
5478 020160 012706 001200          JMP    @NEXT          ;GO TO NEXT TEST...
5479 020164 000177 161252          MCT21:
5480 020170          21$:
5481 020170          MOVE    # 0,BREG          ;SET TO CLEAR SPAD 16...
5482          MICPC=MICPC+1
5483          .WORD .S.
5484 020170 000400          MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS PURPOSES.
5485 020172          MICPC=MICPC+1
5486          .WORD .S.
5487 020172 063236          SSPFLT
5488 020174          15:
5489 020174          1,15,25,35,45,5
5490          MOVE    # 200,BREG          ;START WITH BIT 7.
5491 020174 000002 000600          MICPC=MICPC+1
5491          .WORD .S.

```

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
020202 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 $.
$IFEQ BREG,SPAD <0> 3$ ;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 #4,BREG ;TYPE OF ERROR...
MICPC=MICPC+1
.WORD $.
MOVE BREG,OUT1 <CSR3> ;
MICPC=MICPC+1
.WORD $.
MOVE #5,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 $.
SBR 2$ ;LOOP ON ERROR...
MICPC=MICPC+1
.WORD $.
CALL $ 91 ;IS LOOP DATA SET???
```

55700
55701
55702
55703
55704
55705
55706
55707
55708
55709
55710
55711
55712
55713
55714
55715
55716
55717
55718
55719
55720
55721
55722
55723
55724
55725
55726
55727
55728
55729
55730
55731
55732
55733
55734
55735
55736
55737
55738
55739
55740
55741
55742
55743
55744
55745
55746
55747
55748
55749
55750
55751
55752
55753
55754
55755
55756
55757
55758
55759
55760

020240 000024
020240 000426
020242 000025
020242 104427
020242 000026
020244 040620
020244 000027
020246 100403
020250 000030
020250 040620
020252 000031
020252 061620
020254 000032
020254 103434
020256 000033
020256 100403
020258 000034
020258 000677
020260 000035
020262 063225
020264 000036
020264 062605
020266 000037
020266 043220
020270 000040
020270 062620
020272 000041
020272 060360
020274 000042
020274 101456
020276 000043
020276 061224
020300 000044
020300 060600

```
MOVE # <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 25 ;LOOP ON DATA.  
MOVE MEM,BREG ;RESTORE BREG...  
NICPC=NICPC+1  
.WORD .S.  
SFBRT ;SET THE NEXT BIT...  
NICPC=NICPC+1  
BZ 45 SBR!..SELB!..DBRSH ;BRANCH IF DONE...  
NICPC=NICPC+1  
SBR 25 ;CONTINUE...  
NICPC=NICPC+1  
.WORD .S.  
45:  
SSPFLT 0,115,125,135,145,5  
115: MOVE # 177,BREG ;START WITH BIT 7.  
NICPC=NICPC+1  
.WORD .S.  
125: MOVE BREG,SPAD <5> ;LOAD THE SCRATCH PAD...  
NICPC=NICPC+1  
.WORD .S.  
MOVE SPAD <5>,MEM ;GET THE "FOUND"...  
NICPC=NICPC+1  
.WORD .S.  
MOVE MEM,SPAD <0> ;  
NICPC=NICPC+1  
.WORD .S.  
MOVE BREG,MEM ;SAVE THE CONTENTS OF BREG...  
NICPC=NICPC+1  
$IFEQ BREG,SPAD <0> 135 ;IF GOOD.. CONTINUE...  
SUBC SPAD <0>,BREG,NOP  
NICPC=NICPC+1  
.WORD .S.  
BZ 135  
NICPC=NICPC+1  
.WORD .S.  
MOVE BREG,OUT1 <CSR4> ;ELSE, REPORT ERROR...  
NICPC=NICPC+1  
.WORD .S.  
MOVE SPAD <0>,BREG ;  
NICPC=NICPC+1  
.WORD .S.
```

KMC11 SCRATCH PAD TESTS

5604	020302	000045	MOVE BREG OUT1 (CSRS)	;BAD DATA...
5605		000045	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	020342		SBR SCPE	
5659	020344			

135:

145:

```

5660          000066          MICPC=MICPC+1
5661 020344 104454          .WORD .S.
5662 020346          SBR 21$
5663          000067          MICPC=MICPC+1
5664 020346 170400          .WORD .S.
5665 020350          $SPTS1 6
5666 020350          $XZ
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          $XZ
5674          ;*****
5675
5676 020350          $TSTN
5677          ; TEST 22
5678          ;-----
5679 020350 012737 000022 001202 $TST2: MOV #22,$TSTNM          ; 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          ;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 21$:
5690          MOVE #0,BREG          ;SET TO CLEAR SPAD 16...
5691 020404 000400 MICPC=MICPC+1
5692 020406          .WORD .S.
5693          MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES.
5694          MICPC=MICPC+1
5695          .WORD .S.
5696 $SPFLT 1$:
5697 020410          .WORD .S.
5698          MOVE #200,BREG ;START WITH BIT 7.
5699          MICPC=MICPC+1
5700          .WORD .S.
5701 2$:
5702 020412          MOVE BREG,SPAD <6> ;LOAD THE SCRATCH PAD...
5703          MICPC=MICPC+1
5704          .WORD .S.
5705          MOVE SPAD <6>,MEM ;GET THE "FOUND"...
5706          MICPC=MICPC+1
5707          .WORD .S.
5708          MOVE MEM,SPAD <0> ;
5709          MICPC=MICPC+1
5710          .WORD .S.
5711          MOVE BREG,MEM          ;SAVE THE CONTENTS OF BREG...
5712          MICPC=MICPC+1
5713          .WORD .S.
5714          SIFEQ BREG,SPAD <0> 3$ ;IF GOOD.. CONTINUE...
5715          .WORD .S.
5716          SUB2C SPAD <0>,BREG,NOP
5717          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	ENOR	;DATA ERROR!!
5742	020444		MOVE	B <NICPC+3>,BREG	
5743		000020	NICPC=NICPC+1		
5744	020444	000422	.WORD	.S.	
5745	020446		SBR	ENOR	
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	EN	;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	EN	;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		SHFBT		;SET THE NEXT BIT...
5771		000031	NICPC=NICPC+1		

35:

D2009 MACY11 27(1006) 12-MAY-77 14:07 PAGE 110
D200A.P11 12-MAY-77 13:58

MIC11 SCRATCH PAD TESTS

```

7778 020456 061620 .WORD SBR!..SELB!..DBRSH ;BRANCH IF DONE...
7779 020470 000032 NICPC=NICPC+1
7780 020470 100404 .WORD S. ;CONTINUE...
7781 020472 000032 SBR!..SELB!..DBRSH
7782 020472 100402 NICPC=NICPC+1
7783 020472 000032 .WORD S.
7784 020474 000032 45: 0 115,125,135,145,6
7785 020474 000032 SSFLT 8 177,BREG ;START WITH BIT 7.
7786 020474 000032 119: NICPC=NICPC+1
7787 020474 000032 .WORD S.
7788 020476 000032 125: MOVE BREG,SPAD <6> ;LOAD THE SCRATCH PAD...
7789 020476 000032 NICPC=NICPC+1
7790 020476 000032 .WORD S.
7791 020478 000032 MOVE SPAD <6>,MEM ;GET THE "FOUND"...
7792 020480 000032 NICPC=NICPC+1
7793 020480 000032 .WORD S.
7794 020482 000032 MOVE MEM,SPAD <0> ;
7795 020482 000032 NICPC=NICPC+1
7796 020482 000032 .WORD S.
7797 020484 000032 MOVE BREG,MEM ;SAVE THE CONTENTS OF BREG...
7798 020484 000032 NICPC=NICPC+1
7799 020484 000032 .WORD S.
7800 020486 000032 SIFED BREG,SPAD <0> 135 ;IF GOOD.. CONTINUE...
7801 020486 000032
7802 020506 000041 SUBC SPAD <0>,BREG,NOP
7803 020506 060360 NICPC=NICPC+1
7804 020510 000042 .WORD S.
7805 020510 101456 BZ 135
7806 020510 000042 NICPC=NICPC+1
7807 020512 000042 .WORD S.
7808 020512 101456 MOVE BREG,OUT1 <CSR4> ;ELSE, REPORT ERROR...
7809 020512 000043 NICPC=NICPC+1
7810 020512 061224 .WORD S.
7811 020514 000044 MOVE SPAD <0>,BREG ;
7812 020514 060500 NICPC=NICPC+1
7813 020514 000045 .WORD S.
7814 020516 000045 MOVE BREG,OUT1 <CSR5> ;BAD DATA...
7815 020516 061225 NICPC=NICPC+1
7816 020520 000046 .WORD S.
7817 020520 000404 MOVE BREG,OUT1 <CSR3> ;
7818 020522 000047 NICPC=NICPC+1
7819 020522 061223 .WORD S.
7820 020524 000050 MOVE BREG,OUT1 <CSR7> ;LOAD ADDRESS.
7821 020524 000406 NICPC=NICPC+1
7822 020526 000051 .WORD S.
7823 020526 061227 CALL EROR ;DATA ERROR!!
7824 020530 000051 MOVE B <MICPC+3>,BREG
7825 020530 000051

```

5863
5864
5865
5866
5867
5868
5869
5870
5871
5872
5873
5874
5875
5876
5877
5878
5879
5880
5881
5882
5883

020530 000052
020532 000494

020538 000053
020540 104400

020544 000054
020546 040620

020550 000055
020552 100435
020554

020560 000056
020562 000460

020568 000057
020570 104427
020572

020576 000060
020578 040620
020580

020584 000061
020586 100435
020588

020592 000062
020594 040620
020596

020600 000063
020602 061620
020604

020608 000064
020610 103435
020612
020614
020616

020620 000065
020622 000467
020624

020630 000066
020632 104454

020638 000067
020640 100400
020642
020644

```
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.  
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.  
5SPTS1 7  
5XZ
```

```
***** TEST 23 *****  
* SCRATCH PAD TEST FOR SP7  
* FLOAT A 1 THROUGH SCRATCH PAD 7  
* FLOAT A 0 THROUGH SCRATCH PAD 7  
*****
```

5XZ
5TSTM

KMC11 SCRATCH PAD TESTS

```

; TEST 23
-----
5884 ;
5885 ;
5886 ;
5887 020514 012737 000023 001202 TST23: MOV #23,STSTNM ; LOAD THE NO. OF THIS TEST
5888 020672 012737 021000 001442 MOV #TST24,NEXT ; POINT TO THE START OF NEXT TEST.
5889 ;
5890 ;
5891 020600 004737 035536 JSR PC,LDRVMT ; R1 CONTAINS BASE KMC11 ADDRESS
5892 020604 020620 MCT23 ; LOAD-VERIFY-WAIT.
5893 020606 104022 ERROR 22 ; TIME OUT ERROR...
5894 020610 012706 001200 MOV #STACK,SP ; RESET STACK
5895 020614 000177 160622 JMP @NEXT ; GO TO NEXT TEST...
5896 ;
5897 ;
5898 ;
5899 020620 000000 MOVE #0,BREG ; SET TO CLEAR SPAD 16...
5900 020622 000400 MICPC=MICPC+1
5901 ;
5902 ;
5903 020622 000001 MOVE BREG,SPAD <16> ; FOR RETURN ADDRESS PURPOSES.
5904 020624 063236 MICPC=MICPC+1
5905 ;
5906 ;
5907 020624 000002 SSPFLT 1 15,25,35,45,7
5908 020626 000600 15: MOVE #200,BREG ; START WITH BIT 7.
5909 020628 000003 MICPC=MICPC+1
5910 020630 000600 .WORD $.
5911 020632 063227 25: MOVE BREG,SPAD <7> ; LOAD THE SCRATCH PAD...
5912 020634 000004 MICPC=MICPC+1
5913 020636 043220 .WORD $.
5914 020638 000005 MOVE SPAD <7>,MEM ; GET THE "FOUND"...
5915 020640 062607 MICPC=MICPC+1
5916 020642 043220 .WORD $.
5917 020644 000006 MOVE BREG,MEM ; SAVE THE CONTENTS OF BREG...
5918 020646 062620 MICPC=MICPC+1
5919 020648 000007 SIFEQ BREG,SPAD <0> 35 ; IF GOOD.. CONTINUE...
5920 020650 062620 .WORD $.
5921 020652 000008 SUB2C SPAD <0>,BREG,NOP
5922 020654 060360 MICPC=MICPC+1
5923 020656 000009 .WORD $.
5924 020658 000010 BZ 35
5925 020660 101424 MICPC=MICPC+1
5926 020662 000011 MOVE BREG,OUT1 <CSR4> ; ELSE, REPORT ERROR...
5927 020664 061224 MICPC=MICPC+1
5928 020666 000012 .WORD $.
5929 020668 060600 MOVE SPAD <0>,BREG ;
5930 020670 061224 MICPC=MICPC+1
5931 020672 000013 .WORD $.
5932 020674 060600 MOVE BREG,OUT1 <CSR5> ; BAD DATA...
5933 020676 061225 MICPC=MICPC+1
5934 020678 000014 .WORD $.
5935 020680 000404 MOVE #4,BREG ; TYPE OF ERROR...
5936 020682 000404 MICPC=MICPC+1
5937 020684 000404 .WORD $.
5938 020686 000404 MOVE BREG,OUT1 <CSR3> ;
5939 020688 000404

```

```

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          3S: 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          BR7 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          45: SSPFLT 0,115,125,135,145,7
5987 020710          115: MOVE # 177,BREG ;START WITH BIT 7.
5988 020710          MICPC=MICPC+1
5989          000034      .WORD .S.
5990 020710      000577      .WORD .S.
5991 020712          125: MOVE BREG,SPAD (7) ;LOAD THE SCRATCH PAD...
5992          000035      MICPC=MICPC+1
5993 020712      063227      .WORD .S.
5994 020714          MOVE SPAD (7),MEM ;GET THE "FOUND"...
5995          000036      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 $IFE0 BREG,SPAD <0> 13$ ;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 13$: 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

```

```

6052 020756 104427 .WORD .S.
6053 020760 MOVE MEM BREG ;RESTORE BREG...
6054 000060 MICPC=MICPC+1
6055 020760 040620 .WORD .S.
6056 020762 SBR 12$ ;LOOP ON DATA.
6057 000061 MICPC=MICPC+1
6058 020762 100435 .WORD .S.
6059 020764 MOVE MEM BREG ;RESTORE BREG...
6060 000062 MICPC=MICPC+1
6061 020764 040620 .WORD .S.
6062 020766 SHFBRT ;SET THE NEXT BIT...
6063 000063 MICPC=MICPC+1
6064 020766 061620 .WORD SBR! .SELB! .DBRSH
6065 020770 BBT 12$ ;BRANCH IF NOT DONE...
6066 000064 MICPC=MICPC+1
6067 020770 103435 .WORD .S.
14$: CALL SCPE
6068 020772 MOVE # <MICPC+3>,BREG
6069 020772 MICPC=MICPC+1
6070 020772 .WORD .S.
6071 000065 SBR SCPE
6072 020772 000467 MICPC=MICPC+1
6073 020774 .WORD .S.
6074 000066 SBR 21$
6075 020774 104454 MICPC=MICPC+1
6076 020776 .WORD .S.
6077 000067 SBR 21$
6078 020776 100400 MICPC=MICPC+1
6079 021000 .WORD .S.
6080 021000 $SPTS1 10
6081 $XZ
6082
6083 ;***** TEST 24 *****
6084 ; * SCRATCH PAD TEST FOR SP10
6085 ; * FLOAT A 1 THROUGH SCRATCH PAD 10
6086 ; * FLOAT A 0 THROUGH SCRATCH PAD 10
6087 021000 $XZ
6088 ;*****
6089
6090 021000 $STSN
6091 ; TEST 24
6092 -----
6093 021000 012737 000024 001202 TST24: MOV #24,$STSN ; LOAD THE NO. OF THIS TEST
6094 021006 012737 021214 001442 MOV #TST25,NEXT ; POINT TO THE START OF NEXT TEST.
6095 ; R1 CONTAINS BASE KMC11 ADDRESS
6096 021014 004737 035536 JSR PC,LDRWRT ;LOAD-VERIFY-WAIT.
6097 021020 021034 MCT24
6098 021022 104022 ERROR 22 ; TIME OUT ERROR...
6099 021024 012706 001200 MOV #STACK,SP ; RESET STACK
6100 021030 000177 160406 JMP @NEXT ; GO TO NEXT TEST...
6101 021034 MCT24:
6102 21$:
6103 021034 MOVE #0,BREG ; SET TO CLEAR SPAD 16...
6104 000000 MICPC=MICPC+1
6105 021034 000400 .WORD .S.
6106 021036 MOVE BREG,SPAD <16> ; FOR RETURN ADDRESS PURPOSES.
6107 000001 MICPC=MICPC+1

```

KMC11 SCRATCH PAD TESTS

6108	021036	063236	.WORD .S.	
6109	021040		SSPFLT 1 18,25,38,45,10	
6110	021040		15: MOVE # 200,BREG ;START WITH BIT 7.	
6111		000002	NICPC=NICPC+1	
6112	021040	000600	.WORD .S.	
6113	021042		25: MOVE BREG,SPAD <10> ;LOAD THE SCRATCH PAD...	
6114		000003	NICPC=NICPC+1	
6115	021042	063230	.WORD .S.	
6116	021044		MOVE SPAD <10>,MEM ;GET THE "FOUND"...	
6117		000004	NICPC=NICPC+1	
6118	021044	062610	.WORD .S.	
6119	021046		MOVE MEM,SPAD <0> ;	
6120		000005	NICPC=NICPC+1	
6121	021046	043220	.WORD .S.	
6122	021050		MOVE BREG,MEM ;SAVE THE CONTENTS OF BREG...	
6123		000006	NICPC=NICPC+1	
6124	021050	062620	.WORD .S.	
6125	021052		\$IFEQ BREG,SPAD <0> 3\$;IF GOOD.. CONTINUE...	
6126				
6127				
6128	021052		SUBRC SPAD <0>,BREG,NOP	
6129		000007	NICPC=NICPC+1	
6130	021052	060360	.WORD .S.	
6131	021054		BZ 3\$	
6132		000010	NICPC=NICPC+1	
6133	021054	101424	.WORD .S.	
6134	021056		MOVE BREG,OUT1 <CSR4> ;ELSE, REPORT ERROR...	
6135		000011	NICPC=NICPC+1	
6136	021056	061224	.WORD .S.	
6137	021060		MOVE SPAD <0>,BREG ;	
6138		000012	NICPC=NICPC+1	
6139	021060	060600	.WORD .S.	
6140	021062		MOVE BREG,OUT1 <CSR5> ;BAD DATA...	
6141		000013	NICPC=NICPC+1	
6142	021062	061225	.WORD .S.	
6143	021064		MOVE # 4,BREG ;TYPE OF ERROR...	
6144		000014	NICPC=NICPC+1	
6145	021064	000404	.WORD .S.	
6146	021066		MOVE BREG,OUT1 <CSR3> ;	
6147		000015	NICPC=NICPC+1	
6148	021066	061223	.WORD .S.	
6149	021070		MOVE # 10,BREG ;LOAD ADDRESS.	
6150		000016	NICPC=NICPC+1	
6151	021070	000410	.WORD .S.	
6152	021072		MOVE BREG,OUT1 <CSR7> ;	
6153		000017	NICPC=NICPC+1	
6154	021072	061227	.WORD .S.	
6155	021074		CALL EROR ;DATA ERROR!!	
6156	021074		MOVE # <NICPC+3>,BREG	
6157		000020	NICPC=NICPC+1	
6158	021074	000422	.WORD .S.	
6159	021076		SBR EROR	
6160		000021	NICPC=NICPC+1	
6161	021076	104400	.WORD .S.	
6162	021100		MOVE MEM,BREG ;RESTORE BREG...	
6163		000022	NICPC=NICPC+1	

KMC11 SCRATCH PAD TESTS

6164	021100	040620	.WORD .S.	
6165	021102		SBR 25	;LOOP ON ERROR...
6166		000023	MICPC=MICPC+1	
6167	021102	100403	.WORD .S.	
6168	021104		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:	
6194	021124		SSPFLT 0,118,125,135,145,10	
6195	021124		115: MOVE 8 177,BREG ;START WITH BIT 7.	
6196		000034	MICPC=MICPC+1	
6197	021124	000577	.WORD .S.	
6198	021126		125: MOVE BREG,SPAD <10>	;LOAD THE SCRATCH PAD...
6199		000035	MICPC=MICPC+1	
6200	021126	063230	.WORD .S.	
6201	021130		MOVE SPAD <10>,MEM	;GET THE "FOUND"...
6202		000036	MICPC=MICPC+1	
6203	021130	062610	.WORD .S.	
6204	021132		MOVE MEM,SPAD <0>	
6205		000037	MICPC=MICPC+1	
6206	021132	043220	.WORD .S.	
6207	021134		MOVE BREG,MEM	;SAVE THE CONTENTS OF BREG...
6208		000040	MICPC=MICPC+1	
6209	021134	062620	.WORD .S.	
6210	021136		SIFEQ BREG,SPAD <0> 135	;IF GOOD.. CONTINUE...
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...

```

021142 000043 MICPC=MICPC+1
021144 061224 .WORD .S.
021144 000044 MOVE SPAD (0),BREG ;
021146 060600 MICPC=MICPC+1
021146 000045 .WORD .S. ;BAD DATA...
021146 061225 MOVE BREG,OUT1 (CSRS)
021150 000046 MICPC=MICPC+1 ;TYPE OF ERROR...
021150 061225 .WORD .S.
021150 000046 MOVE # 4,BREG
021152 000404 MICPC=MICPC+1
021152 061223 .WORD .S.
021152 000047 MOVE BREG,OUT1 (CSR3) ;
021154 061223 MICPC=MICPC+1
021154 000050 .WORD .S. ;LOAD ADDRESS.
021154 000410 MICPC=MICPC+1
021156 000051 MOVE BREG,OUT1 (CSR7) ;
021156 061227 MICPC=MICPC+1
021160 000052 .WORD .S. ;DATA ERROR!!
021160 000454 CALL EROR
021162 000053 MOVE # (MICPC+3),BREG
021162 061227 MICPC=MICPC+1
021164 000454 .WORD .S.
021164 000053 SBR EROR
021164 104400 MICPC=MICPC+1 ;RESTORE BREG...
021164 000454 .WORD .S.
021166 000054 MOVE MEM,BREG
021166 040620 MICPC=MICPC+1 ;LOOP ON ERROR...
021166 100435 .WORD .S.
021170 000055 CALL SCP1
021170 000460 MOVE # (MICPC+3),BREG ;IS LOOP DATA SET??
021170 061227 MICPC=MICPC+1
021172 000056 .WORD .S.
021172 000460 SBR SCP1
021174 000057 MICPC=MICPC+1 ;RESTORE BREG...
021174 104427 .WORD .S.
021174 000057 MOVE MEM,BREG
021176 000060 MICPC=MICPC+1 ;LOOP ON DATA.
021176 040620 .WORD .S.
021176 100435 MICPC=MICPC+1 ;RESTORE BREG...
021200 000061 .WORD .S.
021200 040620 MOVE MEM,BREG
021202 000062 MICPC=MICPC+1 ;SET THE NEXT BIT...
021202 061620 .WORD .S.
021202 000063 SHFBRT
021204 061620 MICPC=MICPC+1
021204 000064 .WORD SBR!.SELB!.DBRSH ;BRANCH IF NOT DONE...
021204 103435 BB7 125
021206 000064 MICPC=MICPC+1
021206 103435 .WORD .S.

```

135:

145:


```

6332 021266 SIFE0 BREG,SPAD <0> 35 ;IF GOOD.. CONTINUE...
6333
6334 021266 SUB2C SPAD <0>,BREG,NOP
6335 000007 MICPC=MICPC+1
6336 060360 .WORD .S.
6337 021266
6338 021270 .WORD .S.
6339 000010 MICPC=MICPC+1
6340 101424 .WORD .S.
6341 021270 MOVE BREG,OUT1 <CSR4> ;ELSE, REPORT ERROR...
6342 021272 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 MEN,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 MEN,BREG ;RESTORE BREG...
6383 000026 MICPC=MICPC+1
6384 021326 040620 .WORD .S.
6385 000027 ;LOOP ON DATA.
6386 021326 MICPC=MICPC+1
6387 021326 100403 .WORD .S.

```

35:

```

6388 021330 MOVE MEM,BREG ;RESTORE BREG...
6389 000030 NICPC=NICPC+1
6390 021330 040620 .WORD .S.
6391 021332 SHFBRT ;SET THE NEXT BIT...
6392 000031 NICPC=NICPC+1
6393 021332 061620 .WORD SBR!.SELB!.DBRSH
6394 021334 BB7 45 ;BRANCH IF DONE...
6395 000032 NICPC=NICPC+1
6396 021334 103434 .WORD .S.
6397 021336 SBR 25 ;CONTINUE...
6398 000033 NICPC=NICPC+1
6399 021336 100403 .WORD .S.
6400 021340
6401 021340
6402 021340
45:
SSPFLT 0,115,125,135,145,11
115: MOVE #177,BREG ;START WITH BIT 7.
NICPC=NICPC+1
.WORD .S.
125: MOVE BREG,SPAD <11> ;LOAD THE SCRATCH PAD...
NICPC=NICPC+1
.WORD .S.
MOVE SPAD <11>,MEM ;GET THE "FOUND"...
NICPC=NICPC+1
.WORD .S.
MOVE MEM,SPAD <0> ;
NICPC=NICPC+1
.WORD .S.
MOVE BREG,MEM ;SAVE THE CONTENTS OF BREG...
NICPC=NICPC+1
.WORD .S.
$IFEQ BREG,SPAD <0> 135 ;IF GOOD.. CONTINUE...
6419 021352
6420 021352
6421 000041 SUBRC SPAD <0>,BREG,NOP
6422 021352 060360 NICPC=NICPC+1
6423 021354 .WORD .S.
6424 000042 BZ 135
6425 021354 101456 NICPC=NICPC+1
6426 021356 .WORD .S.
6427 000043 MOVE BREG,OUT1 <CSR4> ;ELSE, REPORT ERROR...
6428 021356 061224 NICPC=NICPC+1
6429 021360 .WORD .S.
6430 000044 MOVE SPAD <0>,BREG ;
6431 021360 060600 NICPC=NICPC+1
6432 021362 .WORD .S.
6433 000045 MOVE BREG,OUT1 <CSR5> ;BAD DATA...
6434 021362 061225 NICPC=NICPC+1
6435 021374 .WORD .S.
6436 000046 MOVE #4,BREG ;TYPE OF ERROR...
6437 021364 000404 NICPC=NICPC+1
6438 021366 .WORD .S.
6439 000047 MOVE BREG,OUT1 <CSR3> ;
6440 021366 061223 NICPC=NICPC+1
6441 021370 .WORD .S.
6442 000050 MOVE #11,BREG ;LOAD ADDRESS.
6443 021370 000411 NICPC=NICPC+1
6443 .WORD .S.
    
```

6444 021372
6445 021372 000051
6446 021372 051227
6447 021374
6448 021374
6449 021374 000052
6450 021374 000454
6451 021376
6452 021376 000053
6453 021376 104400
6454 021400
6455 021400 000054
6456 021402 040620
6457 021402
6458 021402 000055
6459 021402 100435
6460 021404
6461 021404
6462 021404 000056
6463 021404 000460
6464 021406
6465 021406 000057
6466 021406 104427
6467 021410
6468 021410 000060
6469 021410 040620
6470 021412
6471 021412 000061
6472 021412 100435
6473 021414
6474 021414 000062
6475 021414 040620
6476 021416
6477 021416 000063
6478 021416 051620
6479 021420
6480 021420 000064
6481 021420 103435
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

```
MOVE BREG OUT1 (CSR7) ;
NICPC=NICPC+1
.WORD .S.
CALL EROR ;DATA ERROR!!
MOVE # (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 SCPI ;IS LOOP DATA SET???
MOVE # (NICPC+3),BREG
NICPC=NICPC+1
.WORD .S.
SBR SCPI
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.
SHFBRT ;SET THE NEXT BIT...
NICPC=NICPC+1
.WORD .SBR!.SELB!.DBRSH
BB7 125 ;BRANCH IF NOT DONE...
NICPC=NICPC+1
.WORD .S.
CALL SCPE
MOVE # (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 $TSTN ; TEST 26
6505 ;-----
6506
6507 021430 012737 000026 001202 TST26: MOV #26,$TSTNM ; LOAD THE NO. OF THIS TEST
6508 021436 012737 021644 001442 MOV #TST27,NEXT ; POINT TO THE START OF NEXT TEST.
6509 ;R1 CONTAINS BASE KMC11 ADDRESS
6510 021444 004737 035536 JSR PC,LDRWIT ;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 JMP @NEXT ;GO TO NEXT TEST...
6515
6516
6517 021464 MOVE #0,BREG ;SET TO CLEAR SPAD 16...
6518 021464 000000 MICPC=MICPC+1
6519 021464 000400 .WORD $.
6520 021466 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES.
6521 021466 000001 MICPC=MICPC+1
6522 021470 063236 .WORD $.
6523
6524 021470 SSPFLT 1,15,25,35,45,12
6525 18: MOVE #200,BREG ;START WITH BIT 7.
6526 021470 000002 MICPC=MICPC+1
6527 021472 000600 .WORD $.
6528 25: MOVE BREG,SPAD <12> ;LOAD THE SCRATCH PAD...
6529 021472 000003 MICPC=MICPC+1
6530 021474 063232 .WORD $.
6531 021474 MOVE SPAD <12>,MEM ;GET THE "FOUND"...
6532 021474 000004 MICPC=MICPC+1
6533 021476 063512 .WORD $.
6534 021476 MOVE MEM,SPAD <0> ;
6535 021476 000005 MICPC=MICPC+1
6536 021500 043220 .WORD $.
6537 021500 MOVE BREG,MEM ;SAVE THE CONTENTS OF BREG...
6538 021500 000006 MICPC=MICPC+1
6539 021502 062620 SIFEQ BREG,SPAD <0> 35 ;IF GOOD.. CONTINUE...
6540
6541 021502 SUB2C SPAD <0>,BREG,NOP
6542 021502 000007 MICPC=MICPC+1
6543 021504 060360 .WORD $.
6544 021504 BZ 35
6545 021504 000010 MICPC=MICPC+1
6546 021506 101424 .WORD $.
6547 021506 MOVE BREG,OUT1 <CSR4> ;ELSE, REPORT ERROR...
6548 021506 000011 MICPC=MICPC+1
6549 021506 061224 .WORD $.
6550 021510 MOVE SPAD <0>,BREG ;
6551 021510 000012 MICPC=MICPC+1
6552 021510 060600 .WORD $.
6553 021512 MOVE BREG,OUT1 <CSRS> ;BAD DATA...
6554 021512 000013 MICPC=MICPC+1
6555

```

KMC11 SCRATCH PAD TESTS

6558	021512	061225	.WORD .S.	
6559	021514		MOVE 8 4,BREG	;TYPE OF ERROR...
6560		000014	NICPC=NICPC+1	
6561	021514	000404	.WORD .S.	
6562	021516		MOVE 8,BREG OUT1 (CSR3)	;
6563		000015	NICPC=NICPC+1	
6564	021516	061223	.WORD .S.	
6565	021520		MOVE 8 12,BREG	;LOAD ADDRESS.
6566		000016	NICPC=NICPC+1	
6567	021522	000412	.WORD .S.	
6568			MOVE 8,BREG OUT1 (CSR7)	;
6569		000017	NICPC=NICPC+1	
6570	021522	061227	.WORD .S.	
6571			CALL EROR	;DATA ERROR!!
6572	021524		MOVE 8 (NICPC+3),BREG	
6573		000020	NICPC=NICPC+1	
6574	021524	000422	.WORD .S.	
6575			SBR EROR	
6576		000021	NICPC=NICPC+1	
6577	021526	104400	.WORD .S.	
6578	021530		MOVE 8,BREG	;RESTORE BREG...
6579		000022	NICPC=NICPC+1	
6580	021532	040620	.WORD .S.	
6581			SBR 25	;LOOP ON ERROR...
6582		000023	NICPC=NICPC+1	
6583	021532	100403	.WORD .S.	
6584	021534		CALL SCP1	;IS LOOP DATA SET???
6585		000024	MOVE 8 (NICPC+3),BREG	
6586	021534	000426	NICPC=NICPC+1	
6587			.WORD .S.	
6588		000025	SBR SCP1	
6589	021536	104427	NICPC=NICPC+1	
6590			.WORD .S.	
6591	021540		MOVE 8,BREG	;RESTORE BREG...
6592		000026	NICPC=NICPC+1	
6593	021542	040620	.WORD .S.	
6594			SBR 25	;LOOP ON DATA.
6595		000027	NICPC=NICPC+1	
6596	021542	100403	.WORD .S.	
6597			MOVE 8,BREG	;RESTORE BREG...
6598	021544		NICPC=NICPC+1	
6599		000030	.WORD .S.	
6600			SHIFT	;SET THE NEXT BIT...
6601	021546	061620	NICPC=NICPC+1	
6602			.WORD .SBR!..SELB!..DBRSH	
6603	021550		BRT 48	;BRANCH IF DONE...
6604		000032	NICPC=NICPC+1	
6605	021550	103434	.WORD .S.	
6606			SBR 25	;CONTINUE...
6607	021552	000033	NICPC=NICPC+1	
6608	021554	100403	.WORD .S.	
6609				
6610	021554	000034	SSPFLT 0,118,128,138,148,12	
6611		000577	118: MOVE 8 177,BREG ;START WITH BIT 7.	
			NICPC=NICPC+1	
			.WORD .S.	

```

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          SUBEC  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 <CSR5> ;BAD DATA...
6640          000045      MICPC=MICPC+1
6641 021576 061225      .WORD  .S.
6642 021600          MOVE  B 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  B 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  B <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      B (<MICPC+3>),BREG
6669          NICPC=NICPC+1
6670 021620 000460  .WORD      .S.
6671 021622          SBR        SCP1
6672          NICPC=NICPC+1
6673 021622 000057  .WORD      .S.
6674 021624 104427          MOVE      NEH,BREG          ;RESTORE BREG...
6675          NICPC=NICPC+1
6676 021624 040620  .WORD      .S.
6677 021626          SBR        125          ;LOOP ON DATA.
6678          NICPC=NICPC+1
6679 021626 100435  .WORD      .S.
6680 021630          MOVE      NEH,BREG          ;RESTORE BREG...
6681          NICPC=NICPC+1
6682 021630 000062  .WORD      .S.
6683 021632          SHFBRT          ;SET THE NEXT BIT...
6684          NICPC=NICPC+1
6685 021632 061620  .WORD      SBR!..SELB!..DBRSH
6686 021634          SBR        125          ;BRANCH IF NOT DONE...
6687          NICPC=NICPC+1
6688 021634 103435  .WORD      .S.
6689 021636          145:
6690 021636          CALL      SCPE
6691 021636          MOVE      B (<MICPC+3>),BREG
6692          NICPC=NICPC+1
6693 021636 000065  .WORD      .S.
6694 021640          SBR        SCPE
6695          NICPC=NICPC+1
6696 021640 000066  .WORD      .S.
6697 021642          SBR        215
6698          NICPC=NICPC+1
6699 021642 100400  .WORD      .S.
6700 021644          $SPTS1 13
6701 021644          $XZ
6702
6703
6704          ;***** TEST 27 *****
6705          ; * SCRATCH PAD TEST FOR SP13
6706          ; * FLOAT A I THROUGH SCRATCH PAD 13
6707          ; * FLOAT A D THROUGH SCRATCH PAD 13
6708 021644          $XZ
6709          ;*****
6710
6711 021644          $TSTN
6712          ; TEST 27
6713          ;-----
6714 021644 012737 000027 001202  TST27:  MOV      #27,$TSTNM          ; 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,LDRVWT          ;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          MCT27:
6723 021700          215:

```

KMC11 SCRATCH PAD TESTS

```

6724 021700          MOVE      # 0, BREG          ;SET TO CLEAR SPAD 16...
6725          MICPC=MICPC+1
6726 021700 000400    .WORD      .S.
6727 021702          MOVE      BREG, SPAD <16> ;FOR RETURN ADDRESS PURPOSES.
6728          MICPC=MICPC+1
6729 021702 000001    .WORD      .S.
6730 021704          SSPFLT 1, 15, 25, 35, 45, 13
6731 021704 063236    15:      MOVE      # 200, BREG ;START WITH BIT 7.
6732          MICPC=MICPC+1
6733 021704 000600    .WORD      .S.
6734 021706 25:      MOVE      BREG, SPAD <13> ;LOAD THE SCRATCH PAD...
6735          MICPC=MICPC+1
6736 021706 000003    .WORD      .S.
6737 021710          MOVE      SPAD <13>, MEM ;GET THE "FOUND"...
6738          MICPC=MICPC+1
6739 021710 000004    .WORD      .S.
6740 021712          MOVE      MEM, SPAD <0> ;
6741          MICPC=MICPC+1
6742 021712 000005    .WORD      .S.
6743 021714          MOVE      BREG, MEM ;SAVE THE CONTENTS OF BREG...
6744          MICPC=MICPC+1
6745 021714 000006    .WORD      .S.
6746 021716          SIFEQ   BREG, SPAD <0> 35 ;IF GOOD.. CONTINUE...
6747
6748
6749 021716          SUB2C   SPAD <0>, BREG, NOP
6750          MICPC=MICPC+1
6751 021716 000007    .WORD      .S.
6752 021720          BZ      35
6753          MICPC=MICPC+1
6754 021720 101424    .WORD      .S.
6755 021722          MOVE      BREG, OUT1 <CSR4> ;ELSE, REPORT ERROR...
6756          MICPC=MICPC+1
6757 021722 000011    .WORD      .S.
6758 021724          MOVE      SPAD <0>, BREG ;
6759          MICPC=MICPC+1
6760 021724 000012    .WORD      .S.
6761 021726          MOVE      BREG, OUT1 <CSR5> ;BAD DATA...
6762          MICPC=MICPC+1
6763          .WORD      .S.
6764          MOVE      # 4, BREG ;TYPE OF ERROR...
6765          MICPC=MICPC+1
6766          .WORD      .S.
6767          MOVE      BREG, OUT1 <CSR3> ;
6768          MICPC=MICPC+1
6769          .WORD      .S.
6770          MOVE      # 13, BREG ;LOAD ADDRESS.
6771          MICPC=MICPC+1
6772          .WORD      .S.
6773 021734          MOVE      BREG, OUT1 <CSR7> ;
6774          MICPC=MICPC+1
6775 021736 000016    .WORD      .S.
6776 021738 061223    .WORD      .S.
6777 021740          CALL    EROR ;DATA ERROR!!
6778          MOVE      # <MICPC+3>, BREG
6779          MICPC=MICPC+1
6780          .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		CALL	SCP1	;IS LOOP DATA SET???
6790	021750		MOVE	# <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		SHIFB		;SET THE NEXT BIT...
6806		000031	MICPC=MICPC+1		
6807	021762	061620	.WORD	SBR!.SELB!.DBRSH	
6808	021764		B67	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:		
6815	021770		SSPFLT	0,115,125,135,145,13	
6816	021770		115:	MOVE # 177,BREG ;START WITH BIT 7.	
6817		000034	MICPC=MICPC+1		
6818	021770	000577	.WORD	.S.	
6819	021772		125:	MOVE BREG,SPAD <13> ;LOAD THE SCRATCH PAD...	
6820		000035	MICPC=MICPC+1		
6821	021772	063233	.WORD	.S.	
6822	021774		MOVE	SPAD <13>,MEM ;GET THE "FOUND"...	
6823		000036	MICPC=MICPC+1		
6824	021774	062613	.WORD	.S.	
6825	021776		MOVE	MEM,SPAD <0> ;	
6826		000037	MICPC=MICPC+1		
6827	021776	043220	.WORD	.S.	
6828	022000		MOVE	BREG,MEM ;SAVE THE CONTENTS OF BREG...	
6829		000040	MICPC=MICPC+1		
6830	022000	062620	.WORD	.S.	
6831	022002		SIFEQ	BREG,SPAD <0> 135 ;IF GOOD.. CONTINUE...	
6832					
6833					
6834	022002		SUB2C	SPAD <0>,BREG,NOP	
6835		000041	MICPC=MICPC+1		

6836	022002	060360	.WORD	.S	
6837	022004		BZ	125	
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

```

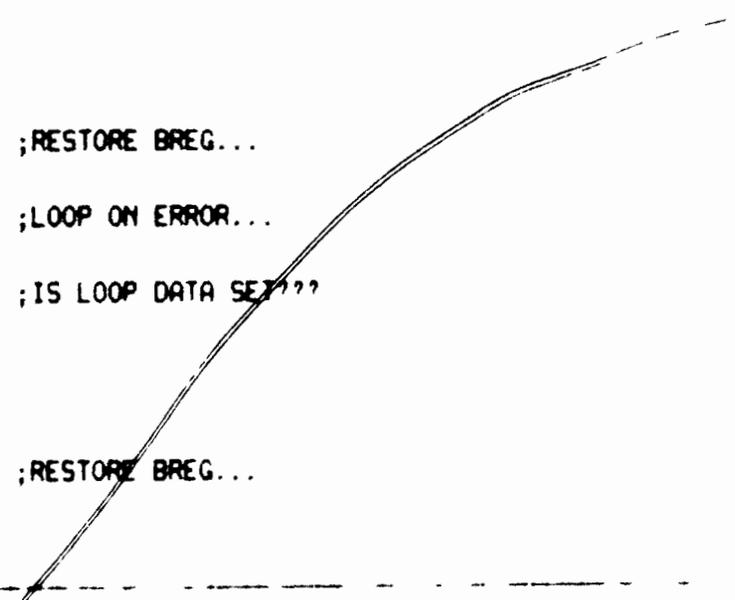
6892 022046 061620 .WORD .SBR!.SELB!.DBRSH
6893 022050 BB7 125 ;BRANCH IF NOT DONE...
6894 000064 MICPC=MICPC+1
6895 103435 .WORD .S.
6896 022050 14S:
6897 022052 CALL SCPE
6898 022052 MOVE # <MICPC+3>,BREG
6899 000065 MICPC=MICPC+1
6900 000467 .WORD .S.
6901 022054 SBR SCPE
6902 000066 MICPC=MICPC+1
6903 022054 104454 .WORD .S.
6904 022056 SBR 21S
6905 000067 MICPC=MICPC+1
6906 022056 100400 .WORD .S.
6907 022060 $SPTS1 14
6908 022060 $XZ
6909
6910
6911 ;***** TEST 30 *****
6912 ;* SCRATCH PAD TEST FOR SP14
6913 ;* FLOAT A 1 THROUGH SCRATCH PAD 14
6914 ;* FLOAT A 0 THROUGH SCRATCH PAD 14
6915 022060 $XZ
6916 ;:*****
6917
6918 $TSTM
6919 ; TEST 30
6920
6921 022060 012737 000030 001202 TST30: MOV #30,$TSTM ; 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,LDRWT ;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 21S:
6931 022114
6932 000000 MOVE #0,BREG ;SET TO CLEAR SPAD 16...
6933 022114 000400 MICPC=MICPC+1
6934 022116 .WORD .S.
6935 000001 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES.
6936 022116 063236 MICPC=MICPC+1
6937 022120 $SPFLT .WORD .S.
6938 022120 1S: 1,1S,2S,3S,4S,14
6939 000002 MOVE #200,BREG ;START WITH BIT 7.
6940 022120 000600 MICPC=MICPC+1
6941 022122 2S: .WORD .S.
6942 000003 MOVE BREG,SPAD <14> ;LOAD THE SCRATCH PAD...
6943 022122 063234 MICPC=MICPC+1
6944 022124 .WORD .S.
6945 000004 MOVE SPAD <14>,MEM ;GET THE "FOUND"...
6946 022124 062614 MICPC=MICPC+1
6947 022126 .WORD .S.
6948 MOVE MEM,SPAD <0> ;

```

KMC11 SCRATCH PAD TESTS

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...

3S:



7004		000026	MICPC=MICPC+1	
7005	022170	040620	.WORD .S.	
7006	022172		SBR 25	; LOOP ON DATA.
7007		000027	MICPC=MICPC+1	
7008	022172	100403	.WORD .S.	
7009	022174		MOVE MEM BREG	; RESTORE BREG...
7010		000030	MICPC=MICPC+1	
7011	022174	040620	.WORD .S.	
7012	022176		SHFBRT	; SET THE NEXT BIT...
7013		000031	MICPC=MICPC+1	
7014	022176	061620	.WORD SBR! .SELB! .DBRSH	
7015	022200		BZ 45	; BRANCH IF DONE...
7016		000032	MICPC=MICPC+1	
7017	022200	103434	.WORD .S.	
7018	022202		SBR 25	; CONTINUE...
7019		000033	MICPC=MICPC+1	
7020	022202	1'0403	.WORD .S.	
7021	022204			
7022	022204			
7023	022204			
7024		000034		
7025	022204	000577		
7026	022206			
7027		000035		
7028	022206	063234		
7029	022210			
7030		000036		
7031	022210	062614		
7032	022212			
7033		000037		
7034	022212	043220		
7035	022214			
7036		000040		
7037	022214	062620		
7038	022216			
7039				
7040				
7041	022216		SUB2C SPAD <0>, BREG, NOP	
7042		000041	MICPC=MICPC+1	
7043	022216	060360	.WORD .S.	
7044	022220		BZ 135	
7045		000042	MICPC=MICPC+1	
7046	022220	101456	.WORD .S.	
7047	022222		MOVE BREG, OUT1 <CSR4>	; ELSE, REPORT ERROR...
7048		000043	MICPC=MICPC+1	
7049	022222	061224	.WORD .S.	
7050	022224		MOVE SPAD <0>, BREG	
7051		000044	MICPC=MICPC+1	
7052	022224	060600	.WORD .S.	
7053	022226		MOVE BREG, OUT1 <CSR5>	; BAD DATA...
7054		000045	MICPC=MICPC+1	
7055	022226	061225	.WORD .S.	
7056	022230		MOVE # 4, BREG	; TYPE OF ERROR...
7057		000046	MICPC=MICPC+1	
7058	022230	000404	.WORD .S.	
7059	022232		MOVE BREG, OUT1 <CSR3>	

45: SSPFLT 0 115, 125, 135, 145, 14
 115: MOVE # 177, BREG ; START WITH BIT 7.

125: MOVE BREG, SPAD <14> ; LOAD THE SCRATCH PAD...

MOVE BREG, MEM ; SAVE THE CONTENTS OF BREG...

SIFEQ BREG, SPAD <0> 135 ; IF GOOD.. CONTINUE...

KMC11 SCRATCH PAD TESTS

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 (CSR?)	;
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 125	;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 125	;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		SFBRT	;SET THE NEXT BIT...
7098		000063	MICPC=MICPC+1	
7099	022262	061620	.WORD SBR!.SELB!.D8RSH	
7100	022264		BRT 125	;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 215	
7112		000067	MICPC=MICPC+1	
7113	022272	100490	.WORD .S.	
7114	022274			
7115	022274			

135:

145:

SSPTS1 15
SXZ

```

7116
7117
7118 ;*****TEST 31 *****
7119 ; SCRATCH PAD TEST FOR SP15
7120 ; FLOAT A 1 THROUGH SCRATCH PAD 15
7121 ; FLOAT A 0 THROUGH SCRATCH PAD 15
7122 022274 SXZ ;*****
7123
7124
7125 022274 STSTM ; TEST 31
7126 ; -----
7127
7128 022274 012737 000031 001202 TST31: MOV #31,STSTM ; LOAD THE NO. OF THIS TEST
7129 022302 012737 022510 001442 MOV #TST32,NEXT ; POINT TO THE START OF NEXT TEST.
7130 ;R1 CONTAINS BASE KMC11 ADDRESS
7131 022310 004737 035536 JSR PC,LDVWRT ;LOAD-VERIFY-WAIT.
7132 022314 022330 MCT31
7133 022316 104022 ERROR 22 ;TIME OUT ERROR...
7134 022320 012706 001200 MOV #STACK,SP ;RESET STACK...
7135 022324 000177 157112 JMP @NEXT ;GO TO NEXT TEST...
7136 022330 MCT31:
7137 022330 215:
7138 022330 MOVE #0,BREG ;SET TO CLEAR SPAD 16...
7139 000000 MICPC=MICPC+1
7140 022330 000400 .WORD $.
7141 022332 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES.
7142 000001 MICPC=MICPC+1
7143 022332 063236 .WORD $.
7144 022334 SSPFLT 1,15,25,35,45,15
7145 022334 15: MOVE #200,BREG ;START WITH BIT 7.
7146 000002 MICPC=MICPC+1
7147 022334 000600 .WORD $.
7148 022336 25: MOVE BREG,SPAD <15> ;LOAD THE SCRATCH PAD...
7149 000003 MICPC=MICPC+1
7150 022336 063235 .WORD $.
7151 022340 MOVE SPAD <15>,MEM ;GET THE "FOUND"...
7152 000004 MICPC=MICPC+1
7153 022340 062615 .WORD $.
7154 022342 MOVE MEM,SPAD <0> ;
7155 000005 MICPC=MICPC+1
7156 022342 043220 .WORD $.
7157 022344 MOVE BREG,MEM ;SAVE THE CONTENTS OF BREG...
7158 000006 MICPC=MICPC+1
7159 022344 062620 .WORD $.
7160 022346 $IFEQ BREG,SPAD <0> 35 ;IF GOOD.. CONTINUE...
7161
7162
7163 022346 SUB2C SPAD <0>,BREG,NOP
7164 000007 MICPC=MICPC+1
7165 022346 060360 .WORD $.
7166 022350 BZ 35
7167 000010 MICPC=MICPC+1
7168 022350 101424 .WORD $.
7169 022352 MOVE BREG,OUT1 <CSR4> ;ELSE, REPORT ERROR...
7170 000011 MICPC=MICPC+1
7171 022352 061224 .WORD $.

```

DZXCA MACY11 27(1006) 13-MAY-77 14:07 PAGE 135
DZXCA.P11 13-MAY-77 13:58 KMC11 SCRATCH PAD TESTS

7172	022354		MOVE SPAD <0>,BREG	;
7173	022354	000012	NICPC=NICPC+1	
7174	022354	060600	.WORD .S.	
7175	022356		MOVE BREG,OUT1 <CSRS>	;BAD DATA...
7176	022356	000013	NICPC=NICPC+1	
7177	022356	061225	.WORD .S.	
7178	022360		MOVE B 4,BREG	;TYPE OF ERROR...
7179	022360	000014	NICPC=NICPC+1	
7180	022360	000404	.WORD .S.	
7181	022362		MOVE BREG,OUT1 <CSR3>	;
7182	022362	000015	NICPC=NICPC+1	
7183	022362	061223	.WORD .S.	
7184	022364		MOVE B 15,BREG	;LOAD ADDRESS.
7185	022364	000016	NICPC=NICPC+1	
7186	022364	000415	.WORD .S.	
7187	022366		MOVE BREG,OUT1 <CSR7>	;
7188	022366	000017	NICPC=NICPC+1	
7189	022366	061227	.WORD .S.	
7190	022370		CALL ENOR	;DATA ERROR!!
7191	022370		MOVE B <NICPC+3>,BREG	
7192	022370	000020	NICPC=NICPC+1	
7193	022370	000422	.WORD .S.	
7194	022372		CALL ENOR	
7195	022372	000021	NICPC=NICPC+1	
7196	022372	104400	.WORD .S.	
7197	022374		MOVE NEH,BREG	;RESTORE BREG...
7198	022374	000022	NICPC=NICPC+1	
7199	022374	040620	.WORD .S.	
7200	022376		CALL ENOR	;LOOP ON ERROR...
7201	022376	000023	NICPC=NICPC+1	
7202	022376	100403	.WORD .S.	
7203	022400		CALL SCP1	;IS LOOP DATA SET???
7204	022400		MOVE B <NICPC+3>,BREG	
7205	022400	000024	NICPC=NICPC+1	
7206	022400	000426	.WORD .S.	
7207	022402		CALL SCP1	
7208	022402	000025	NICPC=NICPC+1	
7209	022402	104427	.WORD .S.	
7210	022404		MOVE NEH,BREG	;RESTORE BREG...
7211	022404	000026	NICPC=NICPC+1	
7212	022404	040620	.WORD .S.	
7213	022406		CALL ENOR	;LOOP ON DATA.
7214	022406	000027	NICPC=NICPC+1	
7215	022406	100403	.WORD .S.	
7216	022410		MOVE NEH,BREG	;RESTORE BREG...
7217	022410	000030	NICPC=NICPC+1	
7218	022410	040620	.WORD .S.	
7219	022412		SHFBRT	;SET THE NEXT BIT...
7220	022412	000031	NICPC=NICPC+1	
7221	022412	061620	.WORD SER!..SELB!..DARSH	
7222	022414		BB7 48	;BRANCH IF DONE...
7223	022414	000032	NICPC=NICPC+1	
7224	022414	103434	.WORD .S.	
7225	022416		SER 28	;CONTINUE...
7226	022416	000033	NICPC=NICPC+1	
7227	022416	100403	.WORD .S.	

35:

KMC11 SCRATCH PAD TESTS

7228	022420		45:	
7229	022420		SSPFLT	0,118,125,135,145,15
7230	022420		118:	MOVE # 177,BREG ;START WITH BIT 7.
7231		000034		MICPC=MICPC+1
7232	022420	000577		.WORD .S.
7233	022422		125:	MOVE BREG,SPAD <15> ;LOAD THE SCRATCH PAD...
7234		000035		MICPC=MICPC+1
7235	022422	063235		.WORD .S.
7236	022424			MOVE SPAD <15>,MEM ;GET THE "FOUND"...
7237		000036		MICPC=MICPC+1
7238	022424	062615		.WORD .S.
7239	022426			MOVE MEM,SPAD <0> ;
7240		000037		MICPC=MICPC+1
7241	022426	043220		.WORD .S.
7242	022430			MOVE BREG,MEM ;SAVE THE CONTENTS OF BREG...
7243		000040		MICPC=MICPC+1
7244	022430	062620		.WORD .S.
7245	022432			\$IFEQ BREG,SPAD <0> 135 ;IF GOOD.. CONTINUE...
7246				
7247				
7248	022432			SUBC SPAD <0>,BREG,NOP
7249		000041		MICPC=MICPC+1
7250	022432	060360		.WORD .S.
7251	022434			BZ 135
7252		000042		MICPC=MICPC+1
7253	022434	101456		.WORD .S.
7254	022436			MOVE BREG,OUT1 <CSR4> ;ELSE, REPORT ERROR...
7255		000043		MICPC=MICPC+1
7256	022436	061224		.WORD .S.
7257	022440			MOVE SPAD <0>,BREG ;
7258		000044		MICPC=MICPC+1
7259	022440	060600		.WORD .S.
7260	022442			MOVE BREG,OUT1 <CSR5> ;BAD DATA...
7261		000045		MICPC=MICPC+1
7262	022442	061225		.WORD .S.
7263	022444			MOVE # 4,BREG ;TYPE OF ERROR...
7264		000046		MICPC=MICPC+1
7265	022444	000404		.WORD .S.
7266	022446			MOVE BREG,OUT1 <CSR3> ;
7267		000047		MICPC=MICPC+1
7268	022446	061223		.WORD .S.
7269	022450			MOVE # 15,BREG ;LOAD ADDRESS.
7270		000050		MICPC=MICPC+1
7271	022450	000415		.WORD .S.
7272	022452			MOVE BREG,OUT1 <CSR7> ;
7273		000051		MICPC=MICPC+1
7274	022452	061227		.WORD .S.
7275	022454			CALL EROR ;DATA ERROR!!
7276	022454			MOVE # <MICPC+3>,BREG
7277		000052		MICPC=MICPC+1
7278	022454	000454		.WORD .S.
7279	022456			SBR EROR
7280		000053		MICPC=MICPC+1
7281	022456	104400		.WORD .S.
7282	022460			MOVE MEM,BREG ;RESTORE BREG...
7283		000054		MICPC=MICPC+1

KMC11 SCRATCH PAD TESTS

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        125
      MICPC=MICPC+1
;LOOP ON ERROR...
135:  .WORD      .S.
      CALL      SCPI
      MOVE      # <MICPC+3>,BREG
;IS LOOP DATA SET??
      MICPC=MICPC+1
      .WORD      .S.
      SBR        125
      MICPC=MICPC+1
      .WORD      .S.
      MOVE      MEM,BREG
;RESTORE BREG...
      MICPC=MICPC+1
      .WORD      .S.
      SBR        125
;LOOP ON DATA.
      MICPC=MICPC+1
      .WORD      .S.
      MOVE      MEM,BREG
;RESTORE BREG...
      MICPC=MICPC+1
      .WORD      .S.
      SHFBRT
;SET THE NEXT BIT...
      MICPC=MICPC+1
      .WORD      .SBR!..SELB!..DBRSH
      BB7       125
;BRANCH IF NOT DCNE...
      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.
SNPR1
SXZ
;***** TEST 32 *****
;* NPR TEST
;* TEST OF DATI, 1 WORD FROM 11 MEMORY TO UPROC
SXZ
;*****
STSTN
; TEST 32
-----
TST32: MOV      #32,STSTNM
      MOV      #ST33,NEXT
; LOAD THE NO. OF THIS TEST
; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
      JSR      PC,LDRNT
      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
JMP @NEXT
MOVE #0,BREG
MICPC=MICPC+1
.MWORD .S.
MOVE BREG,SPAD <16>
MICPC=MICPC+1
.MWORD .S.
MOVE BREG,OUT0 <0>
MICPC=MICPC+1
MOVE BREG,OUT0 <1>
MICPC=MICPC+1
.MWORD .S.
MOVE BREG,OUT0 <4>
MICPC=MICPC+1
.MWORD .S.
MOVE BREG,OUT0 <5>
MICPC=MICPC+1
.MWORD .S.
MOVE BREG,OUT1 <CSR10>
MICPC=MICPC+1
.MWORD .S.
MOVE #125,BREG
MICPC=MICPC+1
.MWORD .S.
MOVE BREG,SPAD <3> ; " " "
MICPC=MICPC+1
.MWORD .S.
SHFBRT
MICPC=MICPC+1
.MWORD SBR! SELB!.DBRSH
MOVE BREG,SPAD <4> ; " " "
.MWORD .S.
MOVE # <NPDAT&377>,BREG ; SET ADDRESS.
MICPC=MICPC+1
.MWORD .S.
MOVE BREG,SPAD <1> ; " " "
MICPC=MICPC+1
.MWORD .S.
MOVE # <NPDAT&177400/400>,BREG ;
MICPC=MICPC+1
.MWORD .S.
MOVE BREG,SPAD <2> ;
MICPC=MICPC+1
.MWORD .S.
MOVE SPAD <1>,BREG ; SET INBA+ADDRESS
MICPC=MICPC+1
.MWORD .S.
MOVE BREG,OUT0 <4> ;
MICPC=MICPC+1
.MWORD .S.
MOVE SPAD <2>,BREG ; SET INBA+ADDRESS
MICPC=MICPC+1

```

```

;RESET STACK...
;GO TO NEXT TEST...
;SET TO CLEAR NPR IBUS REGISTERS.
;FOR RETURN ADDRESS PURPOSE...
;CLEAR IBUS+IN DATA+LB.
;CLEAR IBUS+IN DATA+HB.
;CLEAR INBA+LB.
;CLEAR INBA+HB.
;CLEAR EA+BITS.
;SET GOOD DATA.
; " " "
; SET ADDRESS.
; " " "
;
; SET INBA+ADDRESS
;
; SET INBA+ADDRESS

```

7396	022606	060602	.WORD .S.	
7397	022610		MOVE BREG,OUT0 <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 R0 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 45	;NO, SPIN ON IT??
7410		000026	NICPC=NICPC+1	
7411	022620	102025	.WORD .S.	
7412	022622		MOVE INP0 <0>,BREG	;GET THE DATA LB.
7413		000027	NICPC=NICPC+	
7414	022622	020400	.WORD .S.	
7415	022624		SIFEQ BREG,SPAD <3> 45	;IF GOOD, CHECK NEXT BYTE.
7416				
7417				
7418	022624		SUBRC 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 INP0 <1>,BREG	;GET THE NPR DATA MB.
7450		000042	NICPC=NICPC+1	
7451	022650	020420	.WORD .S.	

```

7452 022652          SIFE0  BREG,SPAD <4>  SS      ;IF GOOD, BRANCH.
7453
7454
7455 022652          SUB2C  SPAD <4>,BREG,NOP
7456 000043          MICPC=MICPC+1
7457 022652 060364    .WORD  .S.
7458 022654          BZ      SS
7459 000044          MICPC=MICPC+1
7460 022654 101454    .WORD  .S.
7461 022656          MOVE   BREG,OUT1 <CSRS>      ;GOOD DATA.
7462 000045          MICPC=MICPC+1
7463 022656 061225    .WORD  .S.
7464 022660          MOVE   SPAD <4>,OUT1 <CSR4>    ;BAD DATA.
7465 000046          MICPC=MICPC+1
7466 022660 061204    .WORD  .S.
7467 022662          MOVE   # 11,MEM          ;TYPE OF ERROR...
7468 000047          MICPC=MICPC+1
7469 022662 002411    .WORD  .S.
7470 022664          MOVE   MEM,OUT1 <CSR3>      ;
7471 000050          MICPC=MICPC+1
7472 022664 041223    .WORD  .S.
7473 022666          CALL   EROR1          ;DATA ERROR...
7474 022666          MOVE   # <MICPC+3>,BREG
7475 000051          MICPC=MICPC+1
7476 022666 000453    .WORD  .S.
7477 022670          SBR   EROR1
7478 000052          MICPC=MICPC+1
7479 022670 104401    .WORD  .S.
7480 022672          SBR   IS          ;LOOP ON ERROR...
7481 000053          MICPC=MICPC+1
7482 022672 100400    .WORD  .S.
7483 022674          SS:   CALL   SCPE          ;SCOPE THE TEST...
7484 022674          MOVE   # <MICPC+3>,BREG
7485 000054          MICPC=MICPC+1
7486 022674 000456    .WORD  .S.
7487 022676          SBR   SCPE
7488 000055          MICPC=MICPC+1
7489 022676 104454    .WORD  .S.
7490 022700          SBR   IS          ;DO THE NEXT ITERATION...
7491 000056          MICPC=MICPC+1
7492 022700 100400    .WORD  .S.
7493 022702          NPDAT: .BYTE 125,252      ;
7494 022704          $XZ   21,WORD,1
7495
7496
7497
7498
7499
7500
7501
7502 022704          $XZ
7503
7504
7505 022704          STSTN
7506
7507

```

```

***** TEST 33 *****
* NPR TEST
* TEST OF DAT0, 1 WORD FROM UPROC TO POP11 MEMORY.
* THEN DAT1 OF THAT WORD AND CHECK....

```

```

*****
;
-----

```

7508	022704	012737	000033	001202	TST33:	MOV	#33,STSTNM	; LOAD THE NO. OF THIS TEST
7509	022712	012737	023154	001442		MOV	#TST34,NEXT	; POINT TO THE START OF NEXT TEST.
7510								;R1 CONTAINS BASE KMC11 ADDRESS
7511	022720	004737	035536			JSR	PC,LDVRWT	;LOAD-VERIFY-WAIT.
7512	022724	022740			MCT33			
7513	022726	104022			ERROR	22		;TIME OUT ERROR...
7514	022730	012706	001200		MOV	#STACK,SP		;RESET STACK...
7515	022734	000177	156502		JMP	#NEXT		;GO TO NEXT TEST...
7516	022740				MCT33:			
7517	022740				IS:	MOVE	#0,BREG	;SET TO CLEAR NPR IBUS+REGISTERS.
7518		000000				MICPC=MICPC+1		
7519	022740	000400				.WORD	.S	
7520	022742					MOVE	BREG,SPAD <16>	;FO RETURN ADDRESS PURPOSE...
7521		000001				MICPC=MICPC+1		
7522	022742	063236				.WORD	.S	
7523	022744					MOVE	BREG,OUT0 <0>	;CLEAR IBUS+IN DATA+LB.
7524		000002				MICPC=MICPC+1		
7525	022744	062220				.WORD	.S	
7526	022746					MOVE	BREG,OUT0 <1>	;CLEAR IBUS+IN DATA+HB.
7527		000003				MICPC=MICPC+1		
7528	022746	062221				.WORD	.S	
7529	022750					MOVE	BREG,OUT0 <2>	;CLEAR IBUS+OUT DATA+LB.
7530		000004				MICPC=MICPC+1		
7531	022750	062222				.WORD	.S	
7532	022752					MOVE	BREG,OUT0 <3>	;CLEAR IBUS+OUT DATA+HB.
7533		000005				MICPC=MICPC+1		
7534	022752	062223				.WORD	.S	
7535	022754					MOVE	BREG,OUT0 <4>	;CLEAR IBUS+IN BA+LB.
7536		000006				MICPC=MICPC+1		
7537	022754	062224				.WORD	.S	
7538	022756					MOVE	BREG,OUT0 <5>	;CLEAR IBUS+IN BA+HB.
7539		000007				MICPC=MICPC+1		
7540	022756	062225				.WORD	.S	
7541	022760					MOVE	BREG,OUT0 <6>	;CLEAR IBUS+OUT BA+LB.
7542		000010				MICPC=MICPC+1		
7543	022760	062226				.WORD	.S	
7544	022762					MOVE	BREG,OUT0 <7>	;CLEAR IBUS+OUT BA+HB.
7545		000011				MICPC=MICPC+1		
7546	022762	062227				.WORD	.S	
7547	022764					MOVE	BREG,OUT1 <CSR10>	;CLEAR IN+BA EABITS.
7548		000012				MICPC=MICPC+1		
7549	022764	061230				.WORD	.S	
7550	022766					MOVE	INP1 <CSR11>,SPAD <0>	;CLEAR OUT+BA EA+BITS.
7551		000013				MICPC=MICPC+1		
7552	022766	123220				.WORD	.S	
7553	022770					OR	SPAD <0>,BREG,OUT1 <CSR11>	;CLEAR OUT+BA EA+BITS.
7554		000014				MICPC=MICPC+1		
7555	022770	061311				.WORD	.S	
7556	022772					MOVE	#125,BREG	; GOOD DATA.
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					SFBRT		
7563		000017				MICPC=MICPC+1		

```

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.
25: 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.
35: 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 ;;

```

```

7620
7621
7622 023044
7623 000042
7624 023044 060601
7625 023046
7626 000043
7627 023046 062224
7628 023050
7629 000044
7630 023050 060602
7631 023052
7632 000045
7633 023052 062225
7634 023054
7635 000046
7636 023054 000401
7637 023056
7638 000047
7639 023056 061230
7640 023060
7641 000050
7642 023060 120600
7643 023062
7644 000051
7645 023062 102050
7646 023064
7647 000052
7648 023064 020400
7649 023066
7650 000053
7651 023066 023040
7652 023070
7653
7654
7655 023070
7656 000054
7657 023070 060360
7658 023072
7659 000055
7660 023072 101466
7661 023074
7662 000056
7663 023074 061224
7664 023076
7665 000057
7666 023076 060600
7667 023100
7668 000060
7669 023100 061225
7670 023102
7671 000061
7672 023102 002411
7673 023104
7674 000062
7675 023104 041223

```

```

::: NOW GET IT BACK BY DATI.
MOVE SPAD <1>,BREG ;SET IN+BA ADDRESS.
MICPC=MICPC+1
.WORD .S.
MOVE BREG,OUTO <4> ; " " " "
MICPC=MICPC+1
.WORD .S.
MOVE SPAD <2>,BREG ; " " " "
MICPC=MICPC+1
.WORD .S.
MOVE BREG,OUTO <5> ; " " " "
MICPC=MICPC+1
.WORD .S.
MOVE #001,BREG ;SET NPR+RQ.
MICPC=MICPC+1
.WORD .S.
MOVE BREG,OUT1 <CSR10> ;SET NPR+IN & NPR+RQ & WRD+XFR.
MICPC=MICPC+1
.WORD .S.
45: MOVE INP1 <CSR10>,BREG ;IS NPR DONE??
MICPC=MICPC+1
.WORD .S.
BBO 45 ;NO, SPIN ON IT...
MICPC=MICPC+1
.WORD .S.
MOVE INPO <0>,BREG ;GET IN DATA+LB.
MICPC=MICPC+1
.WORD .S.
MOVE INPO <2>,SPAD <0> ;BRET OUT+DATA+LB.
MICPC=MICPC+1
.WORD .S.
SIFEQ BREG,SPAD <0> 6S ;BRANCH IF GOOD..

SUB2C SPAD <0>,BREG,NOP
MICPC=MICPC+1
.WORD .S.
BZ 6S
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
.WORD .S.
MOVE BREG,OUT1 <CSR5> ;
MICPC=MICPC+1
.WORD .S.
MOVE #11,MEM ;SET ERROR TYPE...
MICPC=MICPC+1
.WORD .S.
MOVE MEM,OUT1 <CSR3>
MICPC=MICPC+1
.WORD .S.

```

```

7676 023106          CALL  ERROR1 ;DATA ERROR...
7677 023106          MOVE  B (<MICPC+3>),BREG
7678          000063   MICPC=MICPC+1
7679 023106 000465   .WORD  .S.
7680 023110          SBR  ERROR1
7681          000064   MICPC=MICPC+1
7682 023110 104401   .WORD  .S.
7683 023112          SBR  IS ;LOOP ON ERROR...
7684          000065   MICPC=MICPC+1
7685 023112 100400   .WORD  .S.
7686 023114          6S: MOVE  INPO (<1>),BREG ;GET IN DATA + HB....
7687          000066   MICPC=MICPC+1
7688 023114 020420   .WORD  .S.
7689 023116          MOVE  INPO (<3>),SPAD (<0>) ;GET OUT+DATA+HB....
7690          000067   MICPC=MICPC+1
7691 023116 023060   .WORD  .S.
7692 023120          SIFEQ  BREG,SPAD (<0>) 7S ;BRANCH IF GOOD...
7693
7694
7695 023120          SUBRC  SPAD (<0>),BREG,NOP
7696          000070   MICPC=MICPC+1
7697 023120 060360   .WORD  .S.
7698 023122          BZ  7S
7699          000071   MICPC=MICPC+1
7700 023122 101502   .WORD  .S.
7701 023124          MOVE  BREG,OUT1 (<CSR4>) ;GOOD DATA.
7702          000072   MICPC=MICPC+1
7703 023124 061224   .WORD  .S.
7704 023126          MOVE  SPAD (<0>),BREG ;BAD DATA.
7705          000073   MICPC=MICPC+1
7706 023126 060600   .WORD  .S.
7707 023130          MOVE  BREG,OUT1 (<CSR5>) ;
7708          000074   MICPC=MICPC+1
7709 023130 061225   .WORD  .S.
7710 023132          MOVE  B 11 MEM ;SET ERROR TYPE...
7711          000075   MICPC=MICPC+1
7712 023132 002411   .WORD  .S.
7713 023134          MOVE  MEM,OUT1 (<CSR3>) ;
7714          000076   MICPC=MICPC+1
7715 023134 041223   .WORD  .S.
7716 023136          CALL  ERROR1 ;DATA ERROR...
7717 023136          MOVE  B (<MICPC+3>),BREG
7718          000077   MICPC=MICPC+1
7719 023136 000501   .WORD  .S.
7720 023140          SBR  ERROR1
7721          000100   MICPC=MICPC+1
7722 023140 104401   .WORD  .S.
7723 023142          SBR  IS ;LOOP ON ERROR...
7724          000101   MICPC=MICPC+1
7725 023142 100400   .WORD  .S.
7726 023144          7S: CALL  SCPE ;SCOPE THE TEST...
7727 023144          MOVE  B (<MICPC+3>),BREG
7728          000102   MICPC=MICPC+1
7729 023144 000504   .WORD  .S.
7730 023146          SBR  SCPE
7731          000103   MICPC=MICPC+1

```

KMC11 NPR TESTS

7732 023146 104454
 7733 023150
 7734 000104
 7735 023150 100400
 7736 023152 000000
 7737 023154
 7738 023154
 7739
 7740
 7741
 7742
 7743
 7744
 7745 023154
 7746
 7747
 7748 023154
 7749
 7750
 7751 023154 012737 000034 001202
 7752 023162 012737 023322 001442
 7753
 7754 023170 004737 035536
 7755 023174 023210
 7756 023176 104022
 7757 023200 012706 001200
 7758 023204 000177 156232
 7759 023210
 7760 023210
 7761 000000
 7762 023210 000400
 7763 023212
 7764 000001
 7765 023212 063236
 7766 023214
 7767 000002
 7768 023214 063221
 7769 023216
 7770 000003
 7771 023216 062220
 7772 023220
 7773 000004
 7774 023220 062221
 7775 023222
 7776 000005
 7777 023222 062222
 7778 023224
 7779 000006
 7780 023224 062223
 7781 023226
 7782 000007
 7783 023226 062224
 7784 023230
 7785 000010
 7786 023230 062225
 7787 023232

```

      .WORD      .S.
      SBR        IS
      MICPC=MICPC+1
      .WORD      .S.
      .WORD      0
NPRDTI:
SNPR9  21
SXZ

;***** 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
SXZ
;:*****

STSTN
; TEST 34
-----
TST34: MOV      #34,STSTNM      ; LOAD THE NO. OF THIS TEST
      MOV      #ST35,NEXT    ; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
      JSR      PC,LDRVMT     ;LOAD-VERIFY-WAIT.
      MCT34
      ERROR    22           ;TIME OUT ERROR...
      MOV      #STACK,SP    ;RESET STACK...
      JMP      @NEXT        ;GO TO NEXT TEST...

MCT34: IS:
      MOVE     #0,BREG      ;SET TO CLEAR IBUS+REGISTERS.
      MICPC=MICPC+1
      .WORD    .S.
      MOVE     BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSES.
      MICPC=MICPC+1
      .WORD    .S.
      MOVE     BREG,SPAD <1> ;CLEAR DELAY COUNT...
      MICPC=MICPC+1
      .WORD    .S.
      MOVE     BREG,OUT0 <0>
      MICPC=MICPC+1
      .WORD    .S.
      MOVE     BREG,OUT0 <1> ;
      MICPC=MICPC+1
      .WORD    .S.
      MOVE     BREG,OUT0 <2> ;
      MICPC=MICPC+1
      .WORD    .S.
      MOVE     BREG,OUT0 <3> ;
      MICPC=MICPC+1
      .WORD    .S.
      MOVE     BREG,OUT0 <4> ;
      MICPC=MICPC+1
      .WORD    .S.
      MOVE     BREG,OUT0 <5> ;
      MICPC=MICPC+1
      .WORD    .S.
      MOVE     BREG,OUT0 <6> ;
  
```

27(1006) 13-MAY-77 14:07 PAGE 146
13-MAY-77 13:58 KMC11 NPR TESTS

7785	023232	000011	MICPC=MICPC+1	
7790	023234	062226	.WORD .S.	
7791			MOVE BREG,OUT0 <7>	;
7792	023237	000012	MICPC=MICPC+1	
7793	023236	062227	.WORD .S.	
7794			MOVE BREG,OUT1 <CSR10>	;
7795			MICPC=MICPC+1	
7796	023238	000013	.WORD .S.	
7797	023240	061230	25: MOVE # 320,BREG	;NPR ADDRESS+LB.
7798			MICPC=MICPC+1	
7799	023241	000014	.WORD .S.	
7800	023242	000010	MOVE BREG,OUT0 <6>	;LOAD OUT+BA+LB.
7801			MICPC=MICPC+1	
7802	023244	000015	.WORD .S.	
7803	023244	061230	MOVE # 376,BREG	;NPR ADDRESS+HB.
7804			MICPC=MICPC+1	
7805	023246	000016	.WORD .S.	
7806		0610775	MOVE BREG,OUT0 <7>	;LOAD OUT+BA+HB.
7807			MICPC=MICPC+1	
7808	023246	000017	.WORD .S.	
7809	023250	062227	MOVE # 14, MEM	;SET OUT+BA BITS 16 & 17.
7810			MICPC=MICPC+1	
7811	023250	000020	.WORD .S.	
7812	023252	002414	MOVE INP1 <CSR11>,SPAD <0>	;SET OUT+BA 16+17
7813			MICPC=MICPC+1	
7814	023252	000021	.WORD .S.	
7815	023254	123220	OR MEM,SPAD <0>,OUT1 <CSR11>	;SET OUT+BA 16 & 17.
7816			MICPC=MICPC+1	
7817	023254	000022	.WORD .S.	
7818	023256	041311	MOVE # 21,BREG	;
7819			MICPC=MICPC+1	
7820	023256	000023	.WORD .S.	
7821	023260	000421	MOVE BREG,OUT1 <CSR10>	;SET NPR+RD OUT+NPR,WRD+XFR...
7822			MICPC=MICPC+1	
7823	023260	000024	.WORD .S.	
7824	023262	061230	35: MOVE INP1 <CSR11>,BREG	;CHECK IF NON+EX+MEM SET...
7825			MICPC=MICPC+1	
7826	023262	000025	.WORD .S.	
7827	023264	120620	BRO 45	;YES, ITS WORKING.
7828			MICPC=MICPC+1	
7829	023264	000026	.WORD .S.	
7830	023266	102037	SINC SPAD <1>	;WAIT FOR ATLEAST 1BUSEC.
7831			MICPC=MICPC+1	
7832	023266	000027	.WORD .S.	
7833	023266	063061	.WORD .S.!.DSP	;ENOUGH REPORT ERROR...
7834			BC 65	
7835	023270	000030	MICPC=MICPC+1	
7836			.WORD .S.	
7837	023270	000031	SBR 35	;CHECK AGAIN...
7838	023272	101032	MICPC=MICPC+1	
7839			.WORD .S.	
7840	023272	000032	65: MOVE # 13, MEM	;SET ERROR TYPE.
7841	023274	100425	MICPC=MICPC+1	
7842			.WORD .S.	
7843	023274	000033	MOVE MEM,OUT1 <CSR3>	;
7844			MICPC=MICPC+1	
7845	023276	041223	.WORD .S.	

```

7844 023300 CALL ERROR ;REPORT ERROR...
7845 023300 MOVE # (MICPC+3),BREG
7846 000034 MICPC=MICPC+1
7847 023300 .WORD .S
7848 023302 SBR ERROR
7849 000035 MICPC=MICPC+1
7850 023302 104401 .WORD .S
7851 023304 SBR 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 SBR SCPE
7868 000043 MICPC=MICPC+1
7869 023316 104454 .WORD .S
7870 023320 SBR IS ;DO NEXT ITERATION.
7871 000044 MICPC=MICPC+1
7872 023320 100400 .WORD .S
7873 023322 $NPR4
7874 023322 $XZ
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 $XZ
7883 ;:*****
7884
7885 023322 $TSTN
7886
7887 ; TEST 35
7888 023322 012737 000035 001202 TST35: MOV #35,$TSTN ; LOAD THE NO. OF THIS TEST
7889 023330 012737 023606 001442 MOV #TST36,NEXT ; POINT TO THE START OF NEXT TEST.
7890 ;R1 CONTAINS BASE KMC11 ADDRESS
7891 023336 004737 035536 JSR PC,LDRWT ;LOAD-VERIFY-WAIT.
7892 023342 023356 MCT35
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
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	MICPC=MICPC+1		
7902	023360	003013	.WORD	.S.	
7903	023362		MOVE	# 0, BREG	;SET TO CLEAR NPR IBNS+REGISTERS.
7904		000002	MICPC=MICPC+1		
7905	023362	000400	.WORD	.S.	
7906	023364		MOVE	BREG, SPAD <16>	;FOR RETURN ADDRESS PURPOSES...
7907		000003	MICPC=MICPC+1		
7908	023364	063236	.WORD	.S.	
7909	023366		MOVE	BREG, OUT0 <0>	;CLEAR IBUS
7910		000004	MICPC=MICPC+1		
7911	023366	062220	.WORD	.S.	
7912	023370		MOVE	BREG, OUT0 <1>	;CLEAR IBUS.
7913		000005	MICPC=MICPC+1		
7914	023370	062221	.WORD	.S.	
7915	023372		MOVE	BREG, OUT0 <4>	;CLEAR IBUS.
7916		000006	MICPC=MICPC+1		
7917	023372	062224	.WORD	.S.	
7918	023374		MOVE	BREG, OUT0 <5>	;CLEAR IBUS.
7919		000007	MICPC=MICPC+1		
7920	023374	062225	.WORD	.S.	
7921	023376		MOVE	BREG, OUT1 <CSR10>	;CLEAR ER+BITS
7922		000010	MICPC=MICPC+1		
7923	023376	061230	.WORD	.S.	
7924	023400		MOVE	# <INPDT&377>, BREG	;SET NPR ADDRESS.
7925		000011	MICPC=MICPC+1		
7926	023400	000572	.WORD	.S.	
7927	023402		MOVE	BREG, SPAD <1>	;
7928		000012	MICPC=MICPC+1		
7929	023402	063221	.WORD	.S.	
7930	023404		MOVE	# <INPDT&177400/400>, BREG	;
7931		000013	MICPC=MICPC+1		
7932	023404	000447	.WORD	.S.	
7933	023406		MOVE	BREG, SPAD <2>	;
7934		000014	MICPC=MICPC+1		
7935	023406	063222	.WORD	.S.	
7936	023410		MOVE	# 125, BREG	;GET READY TO LOAD MEMORY.
7937		000015	MICPC=MICPC+1		
7938	023410	000525	.WORD	.S.	
7939	023412		MOVE	# 0, MLR	;
7940		000016	MICPC=MICPC+1		
7941	023412	010000	.WORD	.S.	
7942	023414		MOVE	# 2, NPR	;
7943		000017	MICPC=MICPC+1		
7944	023414	004002	.WORD	.S.	
7945	023416		125: MOVE	BREG, MEM, MARINC	;LOAD THE MEMORY...
7946		000020	MICPC=MICPC+1		
7947	023416	076620	.WORD	.S.	
7948	023420		SHFRT		;SET ALTERNATE 125 OR 252.
7949		000021	MICPC=MICPC+1		
7950	023420	061620	.WORD	.SELB!.SELB!.DBRSH	
7951	023422		SDEC	SPAD <13>	;IS IT DONE???
7952		000022	MICPC=MICPC+1		
7953	023422	063173	.WORD	.S.!.DSP	
7954	023424		BZ	155	;YES, START THE TEST.
7955		000023	MICPC=MICPC+1		

7955	023424	101425	.WORD .S.	
7956	023426		SBR 125	;NO, CONTINUE.
7957		000024	MICPC=MICPC+1	
7958	023428	100420	.WORD .S.	
7959	023430		15S: MOVE # 0,MLR	
7960		000025	MICPC=MICPC+1	
7961	023430	010000	.WORD .S.	
7962	023432		MOVE # 0,NPR	
7963		000026	MICPC=MICPC+1	
7964	023432	004000	.WORD .S.	
7965	023434		2S: MOVE SPAD <1>,BREG	;SET IN BA ADDRESS.
7966		000027	MICPC=MICPC+1	
7967	023434	060601	.WORD .S.	
7968	023436		MOVE BREG,OUTO <4>	;SET IN+BA ADDRESS.
7969		000030	MICPC=MICPC+1	
7970	023436	062224	.WORD .S.	
7971	023440		MOVE SPAD <2>,BREG	;SET IN+BA ADDRESS.
7972		000031	MICPC=MICPC+1	
7973	023440	060602	.WORD .S.	
7974	023442		MOVE BREG,OUTO <5>	;SET IN+BA ADDRESS.
7975		000032	MICPC=MICPC+1	
7976	023442	062225	.WORD .S.	
7977	023444		MOVE # 03,BREG	;SET NOT+LAST+XFR & NPR.R1
7978		000033	MICPC=MICPC+1	
7979	023444	000403	.WORD .S.	
7980	023446		MOVE BREG,OUT1 <CSR10>	;SET NOT+LAST+XFR & NPR R1 & WRB.XFR.
7981		000034	MICPC=MICPC+1	
7982	023446	061230	.WORD .S.	
7983	023450		3S: MOVE INP1 <CSR10>,BREG	;IS NPR DONE??
7984		000035	MICPC=MICPC+1	
7985	023450	120600	.WORD .S.	
7986	023452		BRO 35	;NO, SPIN ON IT
7987		000036	MICPC=MICPC+1	
7988	023452	102035	.WORD .S.	
7989	023454		MOVE INPO <0>,MEM MARINC	
7990		000037	MICPC=MICPC+1	
7991	023454	036400	.WORD .S.	
7992	023456		MOVE INPO <1>,MEM MARINC	
7993		000040	MICPC=MICPC+1	
7994	023456	036420	.WORD .S.	
7995	023460		SDEC SPAD <5>	;IS IT DONE??
7996		000041	MICPC=MICPC+1	
7997	023460	063165	.WORD .S.!.DSP	
7998	023462		BZ 45	;YES, GO CHECK THE DATA.
8000		000042	MICPC=MICPC+1	
8001	023462	101447	.WORD .S.	
8002	023464		SINC SPAD <1>	;UPDATE ADDRESS.
8003		000043	MICPC=MICPC+1	
8004	023464	063061	.WORD .S.!.DSP	
8005	023466		SINC SPAD <1>	
8006		000044	MICPC=MICPC+1	
8007	023466	063061	.WORD .S.!.DSP	
8008	023470		SADC SPAD <2>	;UPDATE ADDRESS.
8009		000045	MICPC=MICPC+1	
8010	023470	063102	.WORD .S.!.DSP	
8011	023472		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      55:      MOVE      MEM,SPAD <0>      ;GET THE NPR DATA.
8041      023516      000060      MICPC=MICPC+1
8042      023516      043220      .WORD      .S.
8043      023520      MOVE      SPAD <1>,MLR      ;SET GOOD DATA ADDRESS.
8044      000061      MICPC=MICPC+1
8045      023520      070201      .WORD      .S.
8046      023522      MOVE      SPAD <2>,NPR      ;SET GOOD DATA ADDRESS.
8047      000062      MICPC=MICPC+1
8048      023522      064202      .WORD      .S.
8049      023524      SIFEQ      MEM,SPAD <0> 6S ;IF GOOD CHECK NEXT...
8050      000063      SUB2C      SPAD <0>,MEM,NOP
8051      023524      040360      MICPC=MICPC+1
8052      023526      .WORD      .S.
8053      000064      BZ      6S
8054      023526      101475      MICPC=MICPC+1
8055      023530      .WORD      .S.
8056      023530      MOVE      MEM,OUT1 <CSR4>      ;GOOD DATA.
8057      000065      MICPC=MICPC+1
8058      023532      .WORD      .S.
8059      023532      MOVE      SPAD <0>,BREG      ;BAD DATA.
8060      000066      MICPC=MICPC+1
8061      023532      041224      .WORD      .S.
8062      023534      MOVE      BREG,OUT1 <CSRS>      ;
8063      000067      MICPC=MICPC+1
8064      023534      060600      .WORD      .S.
8065      023534      061225      .WORD      .S.
8066      023534      061225
8067

```

```

8068 023536 MOVE      8 11 MEM          ;ERROR TYPE...
8069      000070 NICPC=NICPC+1
8070 023536 .WORD      .S.
8071 023640 MOVE      MEM OUT1 (CSR3);
8072      000071 NICPC=NICPC+1
8073 023540 .WORD      .S.
8074 023642 CALL      EROR1          ;DATA ERROR...
8075 023642 MOVE      8 (NICPC+3),BREG
8076      000072 NICPC=NICPC+1
8077 023642 .WORD      .S.
8078 023644 SBR      EROR1
8079      000073 NICPC=NICPC+1
8080 023644 .WORD      .S.
8081 023646 SBR      IS          ;LOOP ON ERROR...
8082      000074 NICPC=NICPC+1
8083 023646 .WORD      .S.
8084 023550 6S:      SDEC      SPAD (13)          ;IS IT DONE??
8085      000075 NICPC=NICPC+1
8086 023550 .WORD      .S.!.DSP
8087 023552 BZ      7S          ;YES, SCOPE IT.
8088      000076 NICPC=NICPC+1
8089 023552 .WORD      .S.
8090 023554 SINC      SPAD (3),,MLRLD ;NPR DATA ADDRESS.
8091      000077 NICPC=NICPC+1
8092 023554 .WORD      .S.!.DSP!MLRLD
8093 023556 SINC      SPAD (1)          ;UPDATE GOOD DATA ADDRESS.
8094      000100 NICPC=NICPC+1
8095 023556 .WORD      .S.!.DSP
8096 023560 SADC      SPAD (2)          ;UPDATA GOOD DATA ADDRESS.
8097      000101 NICPC=NICPC+1
8098 023560 .WORD      .S.!.DSP
8099 023562 SBR      5S          ;CHECK NEXT BYTE.
8100      000102 NICPC=NICPC+1
8101 023562 .WORD      .S.
8102 023564 7S:      CALL      SCPE          ;SCOPE IT.
8103 023564 MOVE      8 (NICPC+3),BREG
8104      000103 NICPC=NICPC+1
8105 023564 .WORD      .S.
8106 023566 SBR      SCPE
8107      000104 NICPC=NICPC+1
8108 023566 .WORD      .S.
8109 023570 SBR      IS          ;DO NEXT ITERATION.
8110      000105 NICPC=NICPC+1
8111 023570 .WORD      .S.
8112 023572 .WORD      125125,125125,125125 ;
8113 023600 .WORD      125125,125125,125125 ;
8114      :MNPDT1:      .BLKW      5
8115 023606 $MEM10
8116 023606 $XZ
8117
8118
8119 ;***** TEST 36 *****
8120 ;* MAIN MEMORY TEST
8121 ;* FLOAT A 0 THROUGH ALL MAIN MEMORY LOCATION...
8122 023606 $XZ
8123 ;*****

```

8124					STSTN		
8125	023606					; TEST 36	
8126							
8127							
8128	023606	012737	000036	001202	TST36:	MOV #36,STSTN	; 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,LDVRMT	;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	023642				SMFLT	D,115,125,135,145,155	
8138	023642					MOVE #0,BREG	;SET TO CLEAR SPAD 16...
8139		000000				NICPC=NICPC+1	
8140	023642	000400				.WORD .S.	
8141	023644					MOVE BREG,SPAD <16>	;FOR RETURN ADDRESS PURPOSES...
8142		000001				NICPC=NICPC+1	
8143	023644	063236				.WORD .S.	
8144	023646				115:	MOVE #177,BREG	;START WITH BIT 7.
8145		000002				NICPC=NICPC+1	
8146	023646	000577				.WORD .S.	
8147	023650					MOVE #0,MLR	;LOAD MAR+LO.
8148		000003				NICPC=NICPC+1	
8149	023650	010000				.WORD .S.	
8150	023652					MOVE #0,MPR	;LOAD MAR+HI.
8151		000004				NICPC=NICPC+1	
8152	023652	004000				.WORD .S.	
8153	023654					MOVE #0,MEH	
8154		000005				NICPC=NICPC+1	
8155	023654	002400				.WORD .S.	
8156	023656					MOVE MEH,SPAD <1>	;SET THE COUNTER...
8157		000006				NICPC=NICPC+1	
8158	023656	043221				.WORD .S.	
8159	023660					MOVE MEH,SPAD <3>	;SET THE COUNTER...
8160		000007				NICPC=NICPC+1	
8161	023660	043223				.WORD .S.	
8162							
8163	023662				125:	MOVE BREG,MEH	;SET THE BIT IN MEMORY.
8164		000010				NICPC=NICPC+1	
8165	023662	062620				.WORD .S.	
8166	023664					MOVE MEH,SPAD <2>	;PUT THE "FOUND" IN SPAD <2>.
8167		000011				NICPC=NICPC+1	
8168	023664	043222				.WORD .S.	
8169	023666					MOVE BREG,SPAD <4>	;SAVE THE BREG.
8170		000012				NICPC=NICPC+1	
8171	023666	063224				.WORD .S.	
8172	023670					\$IFEQ BREG,SPAD <2> 135	;IF GOOD CONTINUE.
8173							
8174							
8175	023670					SUB2C SPAD <2>,BREG,NOP	
8176		000013				NICPC=NICPC+1	
8177	023670	060362				.WORD .S.	
8178	023672					BZ 135	
8179		000014				NICPC=NICPC+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 BREG OUT1 <CSR5>	;BAD DATA...
8185		000016	NICPC=NICPC+1	
8186	023676	041225	.WORD .S.	
8187	023700		MOVE 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 ERROR	;REPORT ERROR...
8200	023710		MOVE B <NICPC+3>,BREG	
8201		000023	NICPC=NICPC+1	
8202	023710	000425	.WORD .S.	
8203	023712		SBR ERROR	
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 125	;LOOP ON ERROR...
8210		000026	NICPC=NICPC+1	
8211	023716	100410	.WORD .S.	
8212	023720		CALL SCP1	;LOOP ON DATA SET??
8213	023720		MOVE B <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 125	;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		125 ;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	

KMC11 MAIN MEMORY TESTS

```

8236 023736 063061 .WORD .S.!.DSP
8237 023740          SBOC SPAD (3)
8238          000037 MICPC=MICPC+1
8239 023740 063103 .WORD .S.!.DSP
8240 023742          MOVE # 4, MEM ;PREPARE TO CHECK DONE...
8241          000040 MICPC=MICPC+1
8242 023742 002404 .WORD .S.
8243 023744          $IFLO MEM, SPAD (3) 15$ ;IS IT DONE???
8244          000041
8245          040363
8246 023744          SUB2C SPAD (3), MEM, NOP
8247          000041 MICPC=MICPC+1
8248 023744 040363 .WORD .S.
8249 023746          BC 15$
8250          000042 MICPC=MICPC+1
8251 023746 101046 .WORD .S.
8252          000043
8253 023750          MOVE SPAD (1), MLR ;LOAD MAR+LO.
8254          000043 MICPC=MICPC+1
8255 023750 070201 .WORD .S.
8256 023752          MOVE SPAD (3), MPR ;LOAD MAR+HI.
8257          000044 MICPC=MICPC+1
8258 023752 064203 .WORD .S.
8259 023754          SBR 12$ ;DO THE NEXT MEMORY LOCATION.
8260          000045 MICPC=MICPC+1
8261 023754 100410 .WORD .S.
8262 023756          15$: CALL SCPE ;SCOPE THE TEST...
8263 023756          MOVE # <MICPC+3>, BREG
8264          000046 MICPC=MICPC+1
8265 023756 000450 .WORD .S.
8266 023760          SBR SCPE
8267          000047 MICPC=MICPC+1
8268 023760 104454 .WORD .S.
8269 023762          SBR 11$
8270          000050 MICPC=MICPC+1
8271 023762 100402 .WORD .S.
8272 023764          $MEM11
8273 023764          $XZ
8274
8275
8276          ;***** TEST 37 *****
8277          ;* MAIN MEMORY TEST
8278          ;* FLOAT A 1 THROUGH ALL MAIN MEMORY LOCATIONS...
8279 023764          $XZ
8280          ;*****
8281
8282 023764          $STSN
8283          ; TEST 37
8284          ;-----
8285 023764 012737 000037 001202 TST37: MOV #37, $STSN ; 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 KMC11 ADDRESS
8288 024000 004737 035536 JSR PC, LDVRWT ;LOAD-VERIFY-WAIT.
8289 024004 024020          MCT37
8290 024006 104022          ERROR 22 ;TIME OUT ERROR...
8291 024010 012706 001200 MOV #STACK, SP ;RESET STACK...

```

KMC11 MAIN MEMORY TESTS

8329	024014	000177	155422	JMP	2NEXT		;GO TO NEXT TEST...
8330	024020			MCT37:			
8331	024020			SMPFLT	1, 15, 25, 35, 45, 55		
8332	024020			MOVE	8 0, BREG		;SET TO CLEAR SPAD 16...
8333	024020	000000		NICPC=	NICPC+1		
8334	024020	000400		.WORD	.S.		
8335	024022			MOVE	BREG, SPAD <16>		;FOR RETURN ADDRESS PURPOSES...
8336	024022	000001		NICPC=	NICPC+1		
8337	024022	063236		.WORD	.S.		
8338	024024			15: MOVE	8 200, BREG		;START WITH BIT 7.
8339	024024	000002		NICPC=	NICPC+1		
8340	024024	000600		.WORD	.S.		
8341	024026			MOVE	8 0, MLR		;LOAD MAR+LO.
8342	024026	000003		NICPC=	NICPC+1		
8343	024026	010000		.WORD	.S.		
8344	024030			MOVE	8 0, MPR		;LOAD MAR+HI.
8345	024030	000004		NICPC=	NICPC+1		
8346	024030	004000		.WORD	.S.		
8347	024032			MOVE	8 0, MEM		
8348	024032	000005		NICPC=	NICPC+1		
8349	024032	002400		.WORD	.S.		
8350	024034			MOVE	MEM, SPAD <1>		;SET THE COUNTER...
8351	024034	000006		NICPC=	NICPC+1		
8352	024034	043221		.WORD	.S.		
8353	024036			MOVE	MEM, SPAD <3>		;SET THE COUNTER...
8354	024036	000007		NICPC=	NICPC+1		
8355	024036	043223		.WORD	.S.		
8356	024040			25: MOVE	BREG, MEM		;SET THE BIT IN MEMORY.
8357	024040	000010		NICPC=	NICPC+1		
8358	024040	062620		.WORD	.S.		
8359	024042			MOVE	MEM, SPAD <2>		;PUT THE "FOUND" IN SPAD <2>.
8360	024042	000011		NICPC=	NICPC+1		
8361	024042	043222		.WORD	.S.		
8362	024044			MOVE	BREG, SPAD <4>		;SAVE THE BREG.
8363	024044	000012		NICPC=	NICPC+1		
8364	024044	063224		.WORD	.S.		
8365	024046			SIFEQ	BREG, SPAD <2> 35		;IF GOOD CONTINUE.
8366	024046			SUB2C	SPAD <2>, BREG, NOP		
8367	024046	000013		NICPC=	NICPC+1		
8368	024046	060362		.WORD	.S.		
8369	024050			BZ	35		
8370	024050	000014		NICPC=	NICPC+1		
8371	024050	101427		.WORD	.S.		
8372	024052			MOVE	BREG, OUT1 <CSR4>		;GOOD DATA...
8373	024052	000015		NICPC=	NICPC+1		
8374	024052	061224		.WORD	.S.		
8375	024054			MOVE	MEM, OUT1 <CSR5>		;BAD DATA...
8376	024054	000016		NICPC=	NICPC+1		
8377	024054	041225		.WORD	.S.		
8378	024056			MOVE	8 6, BREG		;SET TYPE OF ERROR...
8379	024056	000017		NICPC=	NICPC+1		
8380	024056	000406		.WORD	.S.		
8381	024060			MOVE	BREG, OUT1 <CSR3>		

8348		000020	MICPC=MICPC+1		
8349	024060	061223	.WORD .S.		
8350	024062		MOVE SPAD <1>,BREG	;LOAD ADDRESS.	
8351		000021	MICPC=MICPC+1		
8352	024062	060601	.WORD .S.		
8353	024064		MOVE BREG OUT1 <CSR7>	;	
8354		000022	MICPC=MICPC+1		
8355	024064	061227	.WORD .S.		
8356	024066		CALL EROR	;REPORT ERROR...	
8357	024066		MOVE # <MICPC+3>,BREG		
8358		000023	MICPC=MICPC+1		
8359	024066	000425	.WORD .S.		
8360	024070		SBR EROR		
8361		000024	MICPC=MICPC+1		
8362	024070	104400	.WORD .S.		
8363	024072		MOVE SPAD <4>,BREG	;RESTORE BREGISTER...	
8364		000025	MICPC=MICPC+1		
8365	024072	060604	.WORD .S.		
8366	024074		SBR 25	;LOOP ON ERROR...	
8367		000026	MICPC=MICPC+1		
8368	024074	100410	.WORD .S.		
8369	024076		CALL SCP1	;LOOP ON DATA SET??	
8370	024076		MOVE # <MICPC+3>,BREG		
8371		000027	MICPC=MICPC+1		
8372	024076	000431	.WORD .S.		
8373	024100		SBR SCP1		
8374		000030	MICPC=MICPC+1		
8375	024100	104427	.WORD .S.		
8376	024102		MOVE SPAD <4>,BREG	;YES, FIRST RESTORE BREG...	
8377		000031	MICPC=MICPC+1		
8378	024102	060604	.WORD .S.		
8379	024104		SBR 25	;SCOPE THE DATA..	
8380		000032	MICPC=MICPC+1		
8381	024104	100410	.WORD .S.		
8382	024106		MOVE SPAD <4>,BREG	;RESTORE BREG...	
8383		000033	MICPC=MICPC+1		
8384	024106	060604	.WORD .S.		
8385	024110		SHFBT	;RIGHT SHIFT BY 1 BIT.	
8386		000034	MICPC=MICPC+1		
8387	024110	061620	.WORD .SBR!.SELB!.DBRSH		
8388	024112		BB7 45	;BRANCH IF DONE...	
8389		000035	MICPC=MICPC+1		
8390	024112	103437	.WORD .S.		
8391	024114		SBR 25	;ELSE, CONTINUE.	
8392		000036	MICPC=MICPC+1		
8393	024114	100410	.WORD .S.		
8394	024116		SINC SPAD <1>	;INCREMENT ADDRESS.	
8395		000037	MICPC=MICPC+1		
8396	024116	063061	.S.!.DSP	;	
8397	024120		SADC SPAD <3>	;	
8398		000040	MICPC=MICPC+1		
8399	024120	063103	.S.!.DSP		
8400	024122		MOVE # 4, MEM	;PREPARE TO CHECK DONE...	
8401		000041	MICPC=MICPC+1		
8402	024122	002404	.WORD .S.		
8403	024124		SIFLO MEM,SPAD <3>	SS ;IS IT DONE???	

```

0404
0405
0406 024124 000042 SUBPC SPAD <3>,MEM,NOP
0407 040363 MICPC=MICPC+1
0408 024124 040363 .WORD .S.
0409 024126 000043 BC SS
0410 040411 101047 MICPC=MICPC+1
0411 024126 101047 .WORD .S.
0412
0413 024130 MOVE SPAD <1>,MLR ;LOAD MAR+LO.
0414 000044 MICPC=MICPC+1
0415 024130 070201 .WORD .S.
0416 024132 MOVE SPAD <3>,MPR ;LOAD MAR+HI.
0417 000045 MICPC=MICPC+1
0418 024132 064203 .WORD .S.
0419 024134 SBR 2$ ;DO THE NEXT MEMORY LOCATION.
0420 000046 MICPC=MICPC+1
0421 024134 100410 .WORD .S.
0422 SS: 024136 CALL SCPE ;SCOPE THE TEST...
0423 024136 MOVE # <MICPC+3>,BREG
0424 000047 MICPC=MICPC+1
0425 024136 000451 .WORD .S.
0426 024140 SBR SCPE
0427 000050 MICPC=MICPC+1
0428 024140 104454 .WORD .S.
0429 024142 SBR 1$
0430 000051 MICPC=MICPC+1
0431 024142 100402 .WORD .S.
0432
0433 $MEM2
0434 $XZ
0435
0436 ;***** TEST 40 *****
0437 ;* MAIN MEMORY DUAL ADDRESSING TEST
0438 ;* LOAD EACH MEMORY LOCATION WITH IT'S OWN PAGE LESS ADDRESS.
0439 ;* READ BACK EACH LOCATION TO VERIFY CORRECT ADDRESSING.
0440 $XZ
0441 ;*****
0442
0443 $STSN
0444 ; TEST 40
0445 -----
0446 024144 012737 000040 001202 TST40: MOV #40,$STSN ; LOAD THE NO. OF THIS TEST
0447 024152 012737 024362 001442 MOV #TST41,NEXT ; POINT TO THE START OF NEXT TEST.
0448 ;R1 CONTAINS BASE KMC11 ADDRESS
0449 024160 004737 035536 JSR PC,LDRWRT ;LOAD-VERIFY-WAIT.
0450 024164 024200 MCT40
0451 024166 104022 ERROR 22 ;TIME OUT ERROR...
0452 024170 012706 001200 MOV #STACK,SP ;RESET STACK...
0453 024174 000177 155242 JMP @NEXT ;GO TO NEXT TEST...
0454 024200 MCT40:
0455 024200 6$: MOVE #0,BREG ;START AT ADDRESS 0.
0456 000000 MICPC=MICPC+1
0457 024200 000400 .WORD .S.
0458 024202 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS PURPOSE...
0459 000001 MICPC=MICPC+1

```

KMC11 MAIN MEMORY TESTS

0460	024202	063236	.WORD .S	
0461	024204		MOVE # 0,HLR	;LOAD MAR+LO.
0462		000002	MICPC=MICPC+1	
0463	024204	010000	.WORD .S	
0464	024206		MOVE # 0,NPR	;LOAD MAR+HI.
0465		000003	MICPC=MICPC+1	
0466	024206	004000	.WORD .S	
0467	024210		MOVE BREG,SPAD (1)	;SET THE PARALLEL ADDRESS.
0468		000004	MICPC=MICPC+1	
0469	024210	063221	.WORD .S	
0470	024212		MOVE BREG,SPAD (2)	;SET THE PARALLEL ADDRESS.
0471		000005	MICPC=MICPC+1	
0472	024212	063222	.WORD .S	
0473	024214		IS: MOVE SPAD (1),MEM	;SET THE ADDRESS IN MEMORY.
0474		000006	MICPC=MICPC+1	
0475	024214	062601	.WORD .S	
0476	024216		MOVE MEM,BREG	;PUT FOUND IN BREG.
0477		000007	MICPC=MICPC+1	
0478	024216	040620	.WORD .S	
0479	024220		SIFEQ BREG,SPAD (1) 2S	;BRANCH IF GOOD...
0480				
0481				
0482	024220		SUB2C SPAD (1),BREG,NOP	
0483		000010	MICPC=MICPC+1	
0484	024220	060361	.WORD .S	
0485	024222		BZ 2S	
0486		000011	MICPC=MICPC+1	
0487	024222	101424	.WORD .S	
0488	024224		MOVE MEM,OUT1 (CSR5)	;LOAD GOOD DATA...
0489		000012	MICPC=MICPC+1	
0490	024224	041225	.WORD .S	
0491	024226		MOVE SPAD (1),BREG	;LOAD BAD DATA...
0492		000013	MICPC=MICPC+1	
0493	024226	060601	.WORD .S	
0494	024230		MOVE BREG,OUT1 (CSR4)	;
0495		000014	MICPC=MICPC+1	
0496	024230	061224	.WORD .S	
0497	024232		MOVE # 7,BREG	;SET THE ERROR TYPE...
0498		000015	MICPC=MICPC+1	
0499	024232	000407	.WORD .S	
0500	024234		MOVE BREG,OUT1 (CSR3)	;
0501		000016	MICPC=MICPC+1	
0502	024234	061223	.WORD .S	
0503	024236		MOVE SPAD (1),BREG	;LOAD ADDRESS.
0504		000017	MICPC=MICPC+1	
0505	024236	060601	.WORD .S	
0506	024240		MOVE BREG,OUT1 (CSR7)	;
0507		000020	MICPC=MICPC+1	
0508	024240	061227	.WORD .S	
0509	024242		CALL ERROR1	;REPORT ERROR...
0510	024242		MOVE # (MICPC+3),BREG	
0511		000021	MICPC=MICPC+1	
0512	024242	000423	.WORD .S	
0513	024244		SBR ERROR1	
0514		000022	MICPC=MICPC+1	
0515	024244	104401	.WORD .S	

DZKCA MACY11 27(1006) 13-MAY-77 14:07 PAGE 159
DZKCA.P11 13-MAY-77 13:58 KMC11 MAIN MEMORY TESTS

```

8516 024246          SBR      15                ;LOOP ON ERROR...
8517 000023          MICPC=MICPC+1
8518 100406          .WORD    .S
28: 024246          CALL     SCP11                ;SCOPE ON DATA SET?
      024248          MOVE     # <MICPC+3>,BREG
      024250          MICPC=MICPC+1
      024252          .WORD    .S
      024254          SBR      15                ;YES, DO IT...
      024256          MICPC=MICPC+1
      024258          .WORD    .S
      02425A          SINC     SPAD <1>,      ,MLRDL ;INCREMENT THE ADDRESS.
      02425C          .WORD    .S
      02425E          .S...DSP:MLRDL
      024260          SADC     SPAD <2>,      ,MPLRD ;INCREMENT PAGE ADDRESS.
      024262          MICPC=MICPC+1
      024264          .S...DSP:MPLRD
      024266          MOVE     # 4,BREG                ;CHECK IF DONE?
      024268          MICPC=MICPC+1
      02426A          .WORD    .S
      02426C          SIFHI   BREG,SPAD <2>  15      ;NO, THEN CONTINUE...

      02426E          SUBIC   SPAD <2>,BREG,NOP
      024270          MICPC=MICPC+1
      024272          .WORD    .S
      024274          BC      64H
      024276          MICPC=MICPC+1
      024278          .WORD    .S
      02427A          SBR      15
      02427C          MICPC=MICPC+1
      02427E          .WORD    .S
64H: 02427E          MOVE     # 0,BREG                ;PREPARE TO CHECK IT.
38: 024280          MICPC=MICPC+1
      024282          .WORD    .S
      024284          MOVE     # 0,MLR
      024286          MICPC=MICPC+1                ;LOAD MAR+LO.
      024288          .WORD    .S
      02428A          MOVE     # 0,MPLR
      02428C          MICPC=MICPC+1                ;LOAD MAR+HI.
      02428E          .WORD    .S
      024290          MOVE     BREG,SPAD <1>
      024292          MICPC=MICPC+1                ;SET THE PARALLEL ADDRESS.
      024294          .WORD    .S
      024296          MOVE     BREG,SPAD <2>
      024298          MICPC=MICPC+1                ;SET THE PARALLEL ADDRESS:
      02429A          .WORD    .S
      02429C          MOVE     REX,BREG
      02429E          MICPC=MICPC+1                ;PUT "FOUND" IN BREG.
48: 024300          .WORD    .S
      024302          SIFEQ   BREG,SPAD <1>  55      ;BRANCH IF GOOD...
8568 024304          .WORD    .S
8569 040620          SIFEQ   BREG,SPAD <1>  55
8570
8571

```

DZXCA MACY11 27(1006) 13-MAY-77 14:07 PAGE 160
DZXCA.P11 13-MAY-77 13:58

KMC11 MAIN MEMORY TESTS

8572	024306	000043	SUBRC SPAD (1),BREG,NOP	
8573		060361	MICPC=MICPC+1	
8574	024306	060361	.WORD .S.	
8575	024310		BZ .S.	
8576		000044	MICPC=MICPC+1	
8577	024310	101455	.WORD .S.	
8578	024312		MOVE MEM,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	060501	.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 ERROR1	;REPORT ERROR.
8594	024324		MOVE # (MICPC+3),BREG	
8595		000052	MICPC=MICPC+1	
8596	024324	000454	.WORD .S.	
8597	024326		SBR ERROR1	
8598		000053	MICPC=MICPC+1	
8599	024326	104401	.WORD .S.	
8600	024330		SBR 45	;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 45	;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		SACC SPAD (2), ,MPRLD	;INCREMENT MEMORY ADDRESS.
8617		000061	MICPC=MICPC+1	
8618	024342	067102	.WORD .S.!.DSP:MPRLD	
8619	024344		MOVE # 4,BREG	;CHECK IF DONE??
8620		000062	MICPC=MICPC+1	
8621	024344	000404	.WORD .S.	
8622	024346		SIFHI BREG,SPAD (2) 45	;NO, THEN CONTINUE...
8623				
8624				
8625	024346		SUBIC SPAD (2),BREG,NOP	
8626		000063	MICPC=MICPC+1	
8627	024346	060342	.WORD .S.	

MAC111 27(1006) 13-MAY-77 13:58 PAGE 161
MAC11 MAIN MEMORY TESTS

```

8629 024350 007064 BC 658
8630 024350 101066 MICPC=MICPC+1
8631 024352 .WORD .S.
8632 000065 SBR 48
8633 100442 MICPC=MICPC+1
8634 024352 .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
8646 024362 MEM31
8647 024362 SXZ
8648
8649 ;***** TEST 41 *****
8650 ;* MAR TEST.
8651 ;* PERFORM DUAL ADDRESSING TEST.
8652 ;* USING MAR AUTO-INC FEATURE.
8653 024362 SXZ
8654 ;*****
8655
8656 024362 STSTN
8657 ; TEST 41
8658
8659 024362 012737 000041 001202 TST41: MOV #41,STSTNM ; LOAD THE NO. OF THIS TEST
8660 024370 012737 024544 001442 MOV #TST42,NEXT ; POINT TO THE START OF NEXT TEST.
8661 ;R1 CONTAINS BASE MAC11 ADDRESS
8662 024376 004737 035536 JSR PC,LDVRWT ;LOAD-VERIFY-WAIT.
8663 024402 024416 MCT41
8664 024404 104022 ERROR 22 ;TIME OUT ERROR...
8665 024406 012706 001200 MOV #STACK,SP ;RESET STACK...
8666 024412 000177 155024 JMP @NEXT ;GO TO NEXT TEST...
8667 024416 MCT41:
8668 024416 18: 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. ;LOAD MAR+HI.
8677 024424 MOVE #0,NPR
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.

```

KMC11 MAIN MEMORY TESTS

8684		000005	MICPC=MICPC+1	
8685	024430	063222	.WORD .S.	
8686	024432		MOVE # 4, BREG	
8687		000006	MICPC=MICPC+1	
8688	024432	000404	.WORD .S.	
8689	024434		25: MOVE SPAD <1>, MEM, MARINC	;LOAD DATA IN MEMORY.
8690	024434	000007	MICPC=MICPC+1	
8691	024436	076601	.WORD .S.	
8692	024436		SINC SPAD <1>	;UPDATE ADDRESS.
8693		000010	MICPC=MICPC+1	
8694	024436	063221	.S.!.DSP	
8695	024436		35: SADC SPAD <2>	;UPDATE ADDRESS.
8696			C=MICPC+1	
8697			DSP	
8698			II BREG, SPAD <2> 25	;BRANCH IF NOT DONE...
8699				
8700			C SPAD <2>, BREG, NOP	
8701			C=MICPC+1	
8702			D .S.	
8703			645	
8704			C=MICPC+1	
8705			D .S.	
8706			25	
8707			C=MICPC+1	
8708			.WORD .S.	
8709				
8710			MOVE # 0, BREG	;START WITH 0 AGAIN.
8711			MICPC=MICPC+1	
8712			.WORD .S.	
8713			MOVE # 0, MLR	;LOAD MAR+LO.
8714			MICPC=MICPC+1	
8715			.WORD .S.	
8716	024452	010000	MOVE # 0, MPR	;LOAD MAR+HI.
8717	024454		MICPC=MICPC+1	
8718		000017	.WORD .S.	
8719	024454	004000	MOVE BREG, SPAD <1>	;SET PARALLEL ADDRESS.
8720	024456		MICPC=MICPC+1	
8721		000020	.WORD .S.	
8722	024456	063221	MOVE BREG, SPAD <2>	;SET PARALLEL ADDRESS.
8723	024460		MICPC=MICPC+1	
8724		000021	.WORD .S.	
8725	024460	063222	35: MOVE MEM, BREG, MARINC	;GET THE CONTENTS OF MEMORY
8726	024462		MICPC=MICPC+1	
8727		000022	.WORD .S.	
8728	024462	054620	SIFE0 BREG, SPAD <1> 45	;BRANCH IF GOOD.
8729	024464			
8730				
8731				
8732	024464		SUB2C SPAD <1>, BREG, NOP	
8733		000023	MICPC=MICPC+1	
8734	024464	060361	.WORD .S.	
8735	024466		BZ 45	
8736		000024	MICPC=MICPC+1	
8737	024466	101436	.WORD .S.	
8738	024470		MOVE MEM, OUT1 <CSRS>	;GOOD DATA.
8739		000025	MICPC=MICPC+1	

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	# 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	EROR		;REPORT ERROR.
8754	024502		MOVE	# (NICPC+3),BREG		
8755		000032		NICPC=NICPC+1		
8756	024502	000434	.WORD	.S.		
8757	024504		SR	EROR		
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		SR	35		;LOOP ON ERROR.
8764		000035		NICPC=NICPC+1		
8765	024510	100422	.WORD	.S.		
8766	024512		CALL	SCP1		;SCOPE THE DATA...
8767	024512		MOVE	# (NICPC+3),BREG		
8768		000036		NICPC=NICPC+1		
8769	024512	000440	.WORD	.S.		
8770	024514		SR	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		SR	35		;LOOP ON DATA...
8777		000041		NICPC=NICPC+1		
8778	024520	100422	.WORD	.S.		
8779	024522		SINC	SPAD (1)		;UPDATE ADDRESS.
8780		000042		NICPC=NICPC+1		
8781	024522	063061	.WORD	.S!.DSP		
8782	024524		SADC	SPAD (2)		;UPDATE ADDRESS.
8783		000043		NICPC=NICPC+1		
8784	024524	063102	.WORD	.S!.DSP		
8785	024526		MOVE	# 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 024530 101050      .WORD      .S.  
8797 024531      .S.  
8798 000047      MICPC=MICPC+1  
8799 024534 100422      .WORD      .S.  
8800 024536  
8801 024538  
8802 024536  
655: CALL SCPE ;SCOPE THE TEST.  
MOVE 8 (MICPC+3),BREG  
MICPC=MICPC+1  
8803 000050      .WORD      .S.  
8804 024536 000452      .S.  
8805 024540      SCPE  
8806 024540      MICPC=MICPC+1  
8807 000051      .WORD      .S.  
8808 024540 104454      .S.  
8809 024542      SBR 15  
8810 000052      MICPC=MICPC+1  
8811 024540 100400      .WORD      .S.  
8812 024544  
8813 024544  
8814  
8815 ;***** TEST 42 *****  
8816 ;#ALU C BIT TEST  
8817 ;#TEST THAT ADD OF 377 AND 1 WILL SET TAG CBIT.  
8818 ;#THEN CHECK IF C BIT CLEARS  
8819 024544 SXZ  
8820  
8821 ;*****  
8822 024544 STSTN  
8823 ; TEST 42  
8824  
8825 024544 012737 000042 001202 TST42: MOV 842 STSTN ; LOAD THE NO. OF THIS TEST  
8826 024552 012737 024700 001442 MOV 8TST43,NEXT ; POINT TO THE START OF NEXT TEST.  
8827 ;R1 CONTAINS BASE KMC11 ADDRESS  
8828 024560 004737 035536 JSR PC,LDRMT ;LOAD-VERIFY-WAIT.  
8829 024564 024600 MCT42  
8830 024566 104022 ERROR 22 ;TIME OUT ERROR...  
8831 024570 012706 001200 MOV 8STACK,SP ;RESET STACK...  
8832 024574 000177 154642 JMP 8NEXT ;GO TO NEXT TEST...  
8833 024600 MCT42:  
8834 024600 15: MOVE 8 0,BREG ;SET TO CLEAR REQUIRED LOCATIONS.  
8835 000000 MICPC=MICPC+1  
8836 024600 000400 .WORD .S.  
8837 024602 MOVE BREG,SPAD (16) ;FOR RETURN ADDRESS PURPOSES.  
8838 000001 MICPC=MICPC+1  
8839 024602 063236 .WORD .S.  
8840 024604 MOVE 8 0,MLR ;  
8841 000002 MICPC=MICPC+1  
8842 024604 010000 .WORD .S.  
8843 024606 MOVE 8 0,MPR ;  
8844 000003 MICPC=MICPC+1  
8845 024606 004000 .WORD .S.  
8846 024610 MOVE BREG,SPAD (0) ;CLEAR SPAD 0.  
8847 000004 MICPC=MICPC+1  
8848 024610 063220 .WORD .S.  
8849 024612 MOVE BREG,SPAD (1) ;CLEAR SPAD 1.  
8850 000005 MICPC=MICPC+1  
8851 024612 063221 .WORD .S.
```

```

8852 024614 000006 MOVE BREG SPAD (2) ;CLEAR SPAD 2.
8853 024614 063222 MICPC=MICPC+1
8854 024614 .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 $INC SPAD (1) ;ADD 1 TO IT.
8862 000011 MICPC=MICPC+1
8863 024622 063061 .WORD .S.!.DSP
8864 024624 .WORD SADC SPAD (2) ;GET THE CARRY BIT..
8865 000012 MICPC=MICPC+1
8866 024624 063102 .WORD .S.!.DSP
8867 024626 .WORD 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 6S: .WORD MOVE ;GOOD DATA.
8874 024632 .WORD MOVE # 21, MEM ;TYPE OF ERROR.
8875 000015 MICPC=MICPC+1
8876 024632 002421 .WORD .S.
8877 024634 .WORD MOVE MEM, OUT1 (CSR3) ;
8878 000016 MICPC=MICPC+1
8879 024634 041223 .WORD .S.
8880 024636 CALL ERROR1 ;REPORT ERROR...
8881 024636 .WORD MOVE # (MICPC+3), BREG
8882 000017 MICPC=MICPC+1
8883 024636 000421 .WORD .S.
8884 024640 .WORD SBR ERROR1
8885 000020 MICPC=MICPC+1
8886 024640 104401 .WORD .S.
8887 024642 .WORD SBR # ;LOOP ON ERROR...
8888 000021 MICPC=MICPC+1
8889 024642 100400 .WORD .S.
8890 024644 2S: .WORD CALL SCP11 ;SCOPE THE DATA??
8891 024644 .WORD MOVE # (MICPC+3), BREG
8892 000022 MICPC=MICPC+1
8893 024644 000424 .WORD .S.
8894 024646 .WORD SBR SCP11
8895 000023 MICPC=MICPC+1
8896 024646 104430 .WORD .S.
8897 024650 .WORD SBR # ;YES, DO IT...
8898 000024 MICPC=MICPC+1
8899 024650 100400 .WORD .S.
8900 024652 3S: .WORD SOEC SPAD (2) ;RESET THE SPAD (2).
8901 000025 MICPC=MICPC+1
8902 024652 063162 .WORD .S.!.DSP
8903 024654 .WORD MOVE # 377, MEM ;LOAD 377 IN MEMORY.
8904 000026 MICPC=MICPC+1
8905 024654 002777 .WORD .S.
8906 024656 .WORD MOVE MEM, SPAD (1) ;LOAD IN SPAD. (1).
8907 000027 MICPC=MICPC+1
    
```

KMC11 ALU TESTS

```

8908 024656 043221      .WORD      $      .S.      WORD      $
8909 024660      .WORD      $INC      SPAD <1>      ;ADD 1 TO IT.
8910      000030      MICPC=MICPC+1
8911 024660 063061      .WORD      $.!.DSP      ;SUBTRACT 1 OUT OF IT.
8912 024662      .WORD      $DEC      SPAD <1>
8913      000031      MICPC=MICPC+1
8914 024662 063161      .WORD      $.!.DSP
8915 024664      .WORD      $ADC      SPAD <0>      ;GET THE CARRY BIT.
8916      000032      MICPC=MICPC+1
8917 024664 063100      .WORD      $.!.DSP
8918 024666      45:      MOVE      SPAD <0>,BREG      ;WAS IT RESET?
8919      000033      MICPC=MICPC+1
8920 024666 060600      .WORD      $.
8921 024670      .WORD      $BIO      $S      ;NO, REPORT ERROR.
8922      000034      MICPC=MICPC+1
8923 024670 102015      .WORD      $.
8924 024672      .WORD      $CALL      $S      ;SCOPE THE TEST.
8925 024672      .WORD      $MOVE      $ <MICPC+3>,BREG
8926      000035      MICPC=MICPC+1
8927 024672 000437      .WORD      $.
8928 024674      .WORD      $SBR      $S      ;DO THE NEXT ITERATION.
8929      000036      MICPC=MICPC+1
8930 024674 104454      .WORD      $.
8931 024676      .WORD      $SBR      $S
8932      000037      MICPC=MICPC+1
8933 024676 100400      .WORD      $.
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,<B>,0
8935 024700      $XZ
8936
8937
8938
8939
8940
8941
8942
8943
8944
8945 024700      $XZ
8946
8947
8948 024700      $STSTN
8949
8950
8951
8952 024700 012737 000043 001202 TST43: MOV      #43,$STSTNM      ; LOAD THE NO. OF THIS TEST
8953 024706 012737 025226 001442      MOV      #TST44,NEXT      ; POINT TO THE START OF NEXT TEST.
8954
8955
8956 024714 004737 035536      JSR      PC,LDRWRT      ;RI CONTAINS BASE KMC11 ADDRESS
8957 024720 024734      MCT43      ;LOAD-VERIFY-WAIT.
8958 024722 104022      ERROR      22      ;TIME OUT ERROR...
8959 024724 012706 001200      MOV      #STACK,SP      ;RESET STACK...
8960 024730 000177 154506      JMP      @NEXT      ;GO TO NEXT TEST...
8961
8962 024734 000000      MCT43: 15:      MOVE      #0,MLR      ;SET MAR+LO.
8963 024736 010000      .WORD      $.
8964      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		SOEC SPAD <2> ;
8982		000007	MICPC=MICPC+1
8983	024752	063162	.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	025004	016400	MICPC=MICPC+1
9022	025006	000025	.WORD .S.
9023	025006	016777	MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY
9024	025010	000026	MICPC=MICPC+1
9025	025010	016777	.WORD .S.
9026	025012	000027	MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.
9027	025012	016777	MICPC=MICPC+1
9028	025014	000030	.WORD .S.
9029	025014	016777	MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.
9030	025016	000031	MICPC=MICPC+1
9031	025016	016525	.WORD .S.
9032	025020	000032	MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
9033	025020	016525	MICPC=MICPC+1
9034	025022	000033	.WORD .S.
9035	025022	016525	MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
9036	025024	000034	MICPC=MICPC+1
9037	025024	016525	.WORD .S.
9038	025026	000035	MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
9039	025026	016652	MICPC=MICPC+1
9040	025026	016652	.WORD .S.
9041	025026	016652	MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
9042	025026	016652	MICPC=MICPC+1
9043	025026	016652	.WORD .S.
9044	025026	016652	MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
9045	025026	016652	MICPC=MICPC+1
9046	025026	016652	.WORD .S.
9047	025026	016652	MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
9048	025026	016652	MICPC=MICPC+1
9049	025026	016652	.WORD .S.
9050	025030	000036	MOVE # 252, MEM MARINC ;RESULT WITH C BIT SET.
9051	025030	016525	MICPC=MICPC+1
9052	025030	016525	.WORD .S.
9053	025032	000037	MOVE # 252, MEM MARINC ;RESULT WITH C BIT SET.
9054	025032	016652	MICPC=MICPC+1
9055	025032	016652	.WORD .S.
9056	025034	000040	MOVE # 252, MEM MARINC ;RESULT WITH C BIT SET.
9057	025034	016652	MICPC=MICPC+1
9058	025034	016652	.WORD .S.
9059	025036	000041	MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
9060	025036	016525	MICPC=MICPC+1
9061	025040	000042	.WORD .S.
9062	025040	016652	MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY
9063	025040	016652	MICPC=MICPC+1
9064	025040	016652	.WORD .S.
9065	025042	000043	MOVE # 125, MEM MARINC ;RESULT WITH C BIT SET.
9066	025042	016525	MICPC=MICPC+1
9067	025042	016525	.WORD .S.
9068	025044	000044	MOVE # 125, MEM MARINC ;RESULT WITH C BIT SET.
9069	025044	016525	MICPC=MICPC+1
9070	025044	016525	.WORD .S.
9071	025046	000045	MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
9072	025046	016652	MICPC=MICPC+1
9073	025046	016652	.WORD .S.
9074	025050	000046	MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
9075	025050	016652	MICPC=MICPC+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. DO	
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> 3\$;BRANCH IF GOOD.
9104				
9105				
9106	025072		SUB2C SPAD <0>, MEM, NOP	
9107		000057	MICPC=MICPC+1	
9108	025072	040360	.WORD .S.	
9109	025074		BZ 3\$	
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 ERROR	;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		35: CALL SCP1
9144	025122		MOVE 8 <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		65: 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 8 377,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 000111 MOVE # 23, BREG ;ERROR TYPE...
9190 025158 000423 MICPC=MICPC+1
9191 025160 000112 .WORD .S
9192 025162 061223 MOVE BREG, OUT1 <CSR3> ;
9193 025164 000112 MICPC=MICPC+1
9194 025166 061223 .WORD .S
9195 025168 000113 MOVE # 11, BREG ;ALU FUNCTION CODE.
9196 025170 000411 MICPC=MICPC+1
9197 025172 000114 .WORD .S
9198 025174 000411 MOVE BREG, OUT1 <CSR7> ;LOAD IT...
9199 025176 000114 MICPC=MICPC+1
9200 025178 061227 .WORD .S
9201 025180 000115 CALL EROR :REPORT ERROR...
9202 025182 000517 MOVE # <MICPC+3>, BREG
9203 025184 000116 MICPC=MICPC+1
9204 025186 000517 .WORD .S
9205 025188 000116 SBR EROR
9206 025190 104400 MICPC=MICPC+1
9207 025192 000117 .WORD .S
9208 025194 070204 MOVE SPAD <4>, MLR ;RESTORE DATA POINTER.
9209 025196 000117 MICPC=MICPC+1
9210 025198 070204 .WORD .S
9211 025200 000120 SBR 6S ;LOOP ON ERROR...
9212 025202 100477 MICPC=MICPC+1
9213 025204 000121 .WORD .S
9214 025206 000523 CALL SCP1 ;SCOPE THE ERROR...
9215 025208 000121 MOVE # <MICPC+3>, BREG
9216 025210 000523 MICPC=MICPC+1
9217 025212 000122 .WORD .S
9218 025214 104427 SBR SCP1
9219 025216 000122 MICPC=MICPC+1
9220 025218 000123 .WORD .S
9221 025220 070204 MOVE SPAD <4>, MLR ;RESTORE DATA POINTER...
9222 025222 000123 MICPC=MICPC+1
9223 025224 070204 .WORD .S
9224 025226 000124 SBR 6S ;SCOPE THE DATA...
9225 025228 100477 MICPC=MICPC+1
9226 025230 000125 .WORD .S
9227 025232 000404 MOVE # 4, BREG ;UPDATE BACKGROUND POINTER.
9228 025234 077004 MICPC=MICPC+1
9229 025236 000126 SADD BREG, SPAD <4>, MARINC ;ALSO DATA POINTER.
9230 025238 063167 .WORD .S !MARINC!.DO
9231 025240 000127 SDEC SPAD <7> ;IS IT DONE??
9232 025242 063167 .WORD .S !.DSP
9233 025244 000127 BZ 4S ;YES, SCOPE THE TEST.
9234 025246 101532 MICPC=MICPC+1
9235 025248 000130 .WORD .S
9236 025250 100453 SBR 2S ;DO, THE NEXT.
9237 025252 000130 MICPC=MICPC+1
9238 025254 000131 .WORD .S
9239 025256 000131 CALL SCPE ;SCOPE THE TEST...
9240 025258 000131 MOVE # <MICPC+3>, BREG
9241 025260 000131 MICPC=MICPC+1
9242
9243

```

025220 000533
025222 000132
025222 104454
025224 000133
025226 100400
025226
025226
025226 012737 000044 001202
025234 012737 025554 001442
025242 004737 035536
025246 025262
025250 104022
025250 012706 001200
025256 000177 154160
025262
025262 000000
025262 010000
025264
025264 000001
025266 004000
025266 000002
025266 000400
025270
025270 000003
025272 063236
025272 000004
025274 063220
025274 000005
025274 063221
025276 000006
025276 063222
025300 000007

```

WORD .S.
SR .SCPE
NICPC=NICPC+1
WORD .S.
SR .IS
;DO THE NEXT ITERATION...
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

;***** 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,STSTNM ; LOAD THE NO. OF THIS TEST
MOV #TST45,NEXT ; 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.
$DEC SPAD <2> ;
NICPC=NICPC+1

```

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

9356	000032	NICPC=NICPC+1
025376	016525	.WORD .S.
025380		MOVE # -1, MEM MARINC
	000033	NICPC=NICPC+1
025380	016777	.WORD .S.
025382		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.
	000034	NICPC=NICPC+1
025382	016777	.WORD .S.
025384		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
	000035	NICPC=NICPC+1
025384	016652	.WORD .S.
025386		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
	000036	NICPC=NICPC+1
025386	016525	.WORD .S.
025390		MOVE # 252, MEM MARINC
	000037	NICPC=NICPC+1
025390	016652	.WORD .S.
025392		MOVE # 252, MEM MARINC ;RESULT WITH C BIT SET.
	000040	NICPC=NICPC+1
025392	016652	.WORD .S.
025394		MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
	000041	NICPC=NICPC+1
025394	016525	.WORD .S.
025396		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY
	000042	NICPC=NICPC+1
025396	016652	.WORD .S.
025370		MOVE # 125, MEM MARINC
	000043	NICPC=NICPC+1
025370	016525	.WORD .S.
025372		MOVE # 125, MEM MARINC ;RESULT WITH C BIT SET.
	000044	NICPC=NICPC+1
025372	016525	.WORD .S.
025374		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
	000045	NICPC=NICPC+1
025374	016652	.WORD .S.
025376		MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
	000046	NICPC=NICPC+1
025376	016652	.WORD .S.
025400		MOVE # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.
	000047	NICPC=NICPC+1
025400	016777	.WORD .S.
025402		MOVE # -1, MEM MARINC ;RESULT WITH C BIT SET.
	000050	NICPC=NICPC+1
025402	016777	.WORD .S.
025404		MOVE # 7, SPAD <7> ;SET THE COUNT.
	000051	NICPC=NICPC+1
025404	003007	.WORD .S.
	025406	MOVE # 0, MLR ;MAR=0.
	000052	NICPC=NICPC+1
025406	010000	.WORD .S.
	25:	MOVE # 0, BREG ;
	000053	NICPC=NICPC+1
025410	000400	.WORD .S.
025412		SADD SPAD <1>, BREG ;CLEAR C BIT.

```

9412      000054      NICPC=NICPC+1
9413      025412      .WORD      $,00
9414      025414      MOVE      MEM,SPAD <0> MARINC      ;GET THE FIRST OPERAND.
9415      000055      NICPC=NICPC+1
9416      025414      .WORD      $,057220
9417      025416      LORH      MEM,SPAD <0>,BR.SP,MARINC      ;SPAD <0>:=SFUNC.
9418      000056      NICPC=NICPC+1
9419      025416      .WORD      $,057640
9420      025420      SIFED     MEM,SPAD <0> 3S ;BRANCH IF GOOD.

025420      SUBRC     SPAD <0>,MEM,NOP
000057      NICPC=NICPC+1
025420      .WORD      $,040260
025422      BZ      $,3S
000060      NICPC=NICPC+1
025422      .WORD      $,101473
025424      MOVE     MEM,OUT1 <4>      ;GOOD DATA
000061      NICPC=NICPC+1
025424      .WORD      $,041224
025426      MOVE     BREG,OUT1 <5>     ;BAD DATA.
000062      NICPC=NICPC+1
025426      .WORD      $,061225
025428      MOVE     # 15,BREG          ;SET TYPE OF ERROR.
000063      NICPC=NICPC+1
025428      .WORD      $,060415
025430      MOVE     BREG,OUT1 <3>     ;SET TYPE OF ERROR.
000064      NICPC=NICPC+1
025430      .WORD      $,061224
025432      MOVE     # 12,BREG          ;LOAD FUNCTION CODE...
000065      NICPC=NICPC+1
025432      .WORD      $,060412
025434      MOVE     BREG,OUT1 <CSR7>   ;LOAD IT...
000066      NICPC=NICPC+1
025434      .WORD      $,061227
025436      CALL     ENOR                ;ALU A OR NOTB ERROR...
000067      NICPC=NICPC+1
025436      .WORD      $,060471
025438      MOVE     # <NICPC+3>,BREG
000070      NICPC=NICPC+1
025438      .WORD      $,104400
025440      MOVE     SPAD <4>,MLR      ;RESET DATA POINTER...
000071      NICPC=NICPC+1
025440      .WORD      $,070804
025442      SRR      $,3S                ;LOOP ON ERROR...
000072      NICPC=NICPC+1
025442      .WORD      $,100463
3S:      CALL     SCP1
000073      NICPC=NICPC+1
025442      .WORD      $,060475
025444      MOVE     # <NICPC+3>,BREG
000074      NICPC=NICPC+1
025444      .WORD      $,104427
025446      MOVE     SPAD <4>,MLR      ;

```

```

9468 000075      MICPC=MICPC+1
9469 025474 070204      .WORD      $
9470 025476      SBR      25      ; SCOPE THE DATA....
9471 000076      MICPC=MICPC+1
9472 025478 100453      .WORD      $
9473 025480      GS:  MOVE     SPAD <4>,MLR      ;RESET DATA POINTER...
9474 000077      MICPC=MICPC+1
9475 025480 070204      .WORD      $
9476 025482      MOVE     MEM,SPAD <0>,MARINC    ;GET FIRST OPRAND...
9477 000100      MICPC=MICPC+1
9478 025482 057220      .WORD      $
9479 025484      MOVE     # 377,BREG      ;
9480 000101      MICPC=MICPC+1
9481 025484 000777      .WORD      $
9482 025486      SADD     SPAD <2>,BREG      ;SET C BIT...
9483 000102      MICPC=MICPC+1
9484 025486 060402      .WORD      $;! DO
9485 025470      LORN     MEM,SPAD <0>,BR.SP,MARINC    ;SP 0 & BR = A OR NOTB
9486 000103      MICPC=MICPC+1
9487 025470 057640      .WORD      $
9488 025472      MOVE     MEM,SPAD <3>,MARINC    ;DUMMY INSTR, TO MARINC.
9489 000104      MICPC=MICPC+1
9490 025472 057223      .WORD      $
9491 025474      SIFER   MEM,SPAD <0>  9S      ;BR IF GOOD...
9492 025474
9493 000105      SUBC    SPAD <0>,MEM,NOP
9494 025474 040360      MICPC=MICPC+1
9495 025476      .WORD      $
9496 025476      BZ      9S
9497 000106      MICPC=MICPC+1
9498 025476 101521      .WORD      $
9499 025500      MOVE     MEM,OUT1 <CSR4> ;GOOD DATA.
9500 000107      MICPC=MICPC+1
9501 025500 041224      .WORD      $
9502 025502      MOVE     BREG,OUT1 <CSR5>      ;BAD DATA.
9503 000110      MICPC=MICPC+1
9504 025502 061225      .WORD      $
9505 025504      MOVE     # 23,BREG      ;ERROR TYPE...
9506 000111      MICPC=MICPC+1
9507 025504 000423      .WORD      $
9508 025506      MOVE     BREG,OUT1 <CSR3>      ;
9509 000112      MICPC=MICPC+1
9510 025506 061223      .WORD      $
9511 025510      MOVE     # 12,BREG      ;ALU FUNCTION CODE.
9512 000113      MICPC=MICPC+1
9513 025510 000412      .WORD      $
9514 025512      MOVE     BREG,OUT1 <CSR7>      ;LOAD IT...
9515 000114      MICPC=MICPC+1
9516 025512 061227      .WORD      $
9517 025514      CALL     EROR      ;REPORT ERROR...
9518 025514      MOVE     # <MICPC+3>,BREG
9519 000115      MICPC=MICPC+1
9520 025514 000517      .WORD      $
9521 025516      SBR     EROR
9522 000116      MICPC=MICPC+1

```

9524 025516 104400
9525 025520
9526 000117
9527 025520 070204
9528 025522
9529 000120
9530 025522 100477
9531 025524
9532 025524
9533 000121
9534 025526 000523
9535 025526
9536 000122
9537 025528 104427
9538 025530
9539 000123
9540 025530 070204
9541 025532
9542 000124
9543 025532 100477
9544 025534
9545 000125
9546 025534 000404
9547 025536
9548 025536 077004
9549 025540
9550 000126
9551 025540 063167
9552 025542
9553 000127
9554 025542 101532
9555 025544
9556 000130
9557 025544 100453
9558 025546
9559 025546
9560 000131
9561 025546 000533
9562 025550
9563 000132
9564 025550 104454
9565 025552
9566 000133
9567 025552 100400
9568 025554
9569 025554
9570
9571
9572
9573
9574
9575
9576
9577
9578
9579 025554

```
.WORD .S.  
MOVE SPAD <4>,MLR ;RESTORE DATA POINTER.  
MICPC=MICPC+1  
.WORD .S.  
SBR 65 ;LOOP ON ERROR...  
MICPC=MICPC+1  
95: .WORD .S.  
CALL SCP1 ;SCOPE THE ERROR...  
MOVE # <MICPC+3>,BREG  
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 65 ;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 25 ;DO, THE NEXT.  
MICPC=MICPC+1  
45: .WORD .S.  
CALL SCPE ;SCOPE THE TEST...  
MOVE # <MICPC+3>,BREG  
MICPC=MICPC+1  
.WORD .S.  
SBR SCPE  
MICPC=MICPC+1  
.WORD .S.  
SBR 15 ;DO THE NEXT ITERATION...  
MICPC=MICPC+1  
SALUT1 .WORD .S.  
SXZ 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  
  
:***** 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                                     ; R1 CONTAINS BASE KMC11 ADDRESS
9588 025570 004737 035536 JSR PC,LDVRMT          ;LOAD-VERIFY-WAIT.
9589 025574 025610 MCT45
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 025610 MCT45:
9594 025610 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 000008 MOVE BREG,SPAD <4> ;
9619 025630 063124 MICPC=MICPC+1
9620 000009 .WORD .S.
9621 000010 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY.
9622 025630 016400 MICPC=MICPC+1
9623 000011 .WORD .S.
9624 000012 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY
9625 025634 016400 MICPC=MICPC+1
9626 000013 .WORD .S.
9627 025636 MOVE #0, MEM MARINC
9628 000014 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		SUBC 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 <CSRS>	; BAD DATA.
9821		000110	MICPC=MICPC+1	
9822	026030	061225	WORD .S.	
9823	026032		MOVE B 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 B 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 B <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 B <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	

```

9860 026060 100477      .WORD      .S.      ;
9861 026061 000125      .WORD      .S.      ;
9862 026062 000404      .WORD      .S.      ;
9863 026062 000404      .WORD      .S.      ;
9864 026064 077004      .WORD      .S.      ;
9865 026064 077004      .WORD      .S.      ;
9866 026066 000126      .WORD      .S.      ;
9867 026066 063167      .WORD      .S.      ;
9868 026066 063167      .WORD      .S.      ;
9869 026070 000127      .WORD      .S.      ;
9870 026070 101532      .WORD      .S.      ;
9871 026072 000130      .WORD      .S.      ;
9872 026072 100453      .WORD      .S.      ;
9873 026072 100453      .WORD      .S.      ;
9874 026074 000131      .WORD      .S.      ;
9875 026074 000533      .WORD      .S.      ;
9876 026074 000132      .WORD      .S.      ;
9877 026074 104454      .WORD      .S.      ;
9878 026076 000133      .WORD      .S.      ;
9879 026100 100400      .WORD      .S.      ;
9880 026100 100400      .WORD      .S.      ;
9881 026102 000133      .WORD      .S.      ;
9882 026102 100400      .WORD      .S.      ;
9883 026102 100400      .WORD      .S.      ;
9884 026102 100400      .WORD      .S.      ;
9885 026102 100400      .WORD      .S.      ;
9886 026102 100400      .WORD      .S.      ;
9887 026102 100400      .WORD      .S.      ;
9888 026102 100400      .WORD      .S.      ;
9889 026102 100400      .WORD      .S.      ;
9890 026102 100400      .WORD      .S.      ;
9891 026102 100400      .WORD      .S.      ;
9892 026102 100400      .WORD      .S.      ;
9893 026102 100400      .WORD      .S.      ;
9894 026102 100400      .WORD      .S.      ;
9895 026102 100400      .WORD      .S.      ;
9896 026102 100400      .WORD      .S.      ;
9897 026102 100400      .WORD      .S.      ;
9898 026102 100400      .WORD      .S.      ;
9899 026102 100400      .WORD      .S.      ;
9900 026102 100400      .WORD      .S.      ;
9901 026102 100400      .WORD      .S.      ;
9902 026110 012737 000046 001202 TST46: MOV      #46,STSTNM      ; LOAD THE NO. OF THIS TEST
9903 026110 012737 026430 001442      MOV      #TST47,NEXT      ; POINT TO THE START OF NEXT TEST.
9904 026116 004737 035536      JSR      PC,LDRMNT      ;R1 CONTAINS BASE KMC11 ADDRESS
9905 026122 026136      MCT46      ;LOAD-VERIFY-WAIT.
9906 026124 104022      ERROR      22      ;TIME OUT ERROR...
9907 026126 012706 001200      MOV      #STACK,SP      ;RESET STACK...
9908 026132 000177 153304      JMP      @NEXT      ;GO TO NEXT TEST...
9909 026136 000000      MCT46:      ;
9910 026136 010000      15:      MOVE      #0,MLR      ;SET MAR+LO.
9911 026136 000001      .WORD      .S.      ;
9912 026140 000001      MOVE      #0,MPR      ;SET MAR+HI.
9913 026140 000001      .WORD      .S.      ;
9914 026140 000001      .WORD      .S.      ;
9915 026140 000001      .WORD      .S.      ;
9916 026140 000001      .WORD      .S.      ;
9917 026140 000001      .WORD      .S.      ;
9918 026140 000001      .WORD      .S.      ;
9919 026140 000001      .WORD      .S.      ;
9920 026140 000001      .WORD      .S.      ;
9921 026140 000001      .WORD      .S.      ;
9922 026140 000001      .WORD      .S.      ;
9923 026140 000001      .WORD      .S.      ;
9924 026140 000001      .WORD      .S.      ;
9925 026140 000001      .WORD      .S.      ;
9926 026140 000001      .WORD      .S.      ;
9927 026140 000001      .WORD      .S.      ;
9928 026140 000001      .WORD      .S.      ;
9929 026140 000001      .WORD      .S.      ;
9930 026140 000001      .WORD      .S.      ;
9931 026140 000001      .WORD      .S.      ;
9932 026140 000001      .WORD      .S.      ;
9933 026140 000001      .WORD      .S.      ;
9934 026140 000001      .WORD      .S.      ;
9935 026140 000001      .WORD      .S.      ;
9936 026140 000001      .WORD      .S.      ;
9937 026140 000001      .WORD      .S.      ;
9938 026140 000001      .WORD      .S.      ;
9939 026140 000001      .WORD      .S.      ;
9940 026140 000001      .WORD      .S.      ;
9941 026140 000001      .WORD      .S.      ;
9942 026140 000001      .WORD      .S.      ;
9943 026140 000001      .WORD      .S.      ;
9944 026140 000001      .WORD      .S.      ;
9945 026140 000001      .WORD      .S.      ;
9946 026140 000001      .WORD      .S.      ;
9947 026140 000001      .WORD      .S.      ;
9948 026140 000001      .WORD      .S.      ;
9949 026140 000001      .WORD      .S.      ;
9950 026140 000001      .WORD      .S.      ;
9951 026140 000001      .WORD      .S.      ;
9952 026140 000001      .WORD      .S.      ;
9953 026140 000001      .WORD      .S.      ;
9954 026140 000001      .WORD      .S.      ;
9955 026140 000001      .WORD      .S.      ;
9956 026140 000001      .WORD      .S.      ;
9957 026140 000001      .WORD      .S.      ;
9958 026140 000001      .WORD      .S.      ;
9959 026140 000001      .WORD      .S.      ;
9960 026140 000001      .WORD      .S.      ;
9961 026140 000001      .WORD      .S.      ;
9962 026140 000001      .WORD      .S.      ;
9963 026140 000001      .WORD      .S.      ;
9964 026140 000001      .WORD      .S.      ;
9965 026140 000001      .WORD      .S.      ;
9966 026140 000001      .WORD      .S.      ;
9967 026140 000001      .WORD      .S.      ;
9968 026140 000001      .WORD      .S.      ;
9969 026140 000001      .WORD      .S.      ;
9970 026140 000001      .WORD      .S.      ;
9971 026140 000001      .WORD      .S.      ;
9972 026140 000001      .WORD      .S.      ;
9973 026140 000001      .WORD      .S.      ;
9974 026140 000001      .WORD      .S.      ;
9975 026140 000001      .WORD      .S.      ;
9976 026140 000001      .WORD      .S.      ;
9977 026140 000001      .WORD      .S.      ;
9978 026140 000001      .WORD      .S.      ;
9979 026140 000001      .WORD      .S.      ;
9980 026140 000001      .WORD      .S.      ;
9981 026140 000001      .WORD      .S.      ;
9982 026140 000001      .WORD      .S.      ;
9983 026140 000001      .WORD      .S.      ;
9984 026140 000001      .WORD      .S.      ;
9985 026140 000001      .WORD      .S.      ;
9986 026140 000001      .WORD      .S.      ;
9987 026140 000001      .WORD      .S.      ;
9988 026140 000001      .WORD      .S.      ;
9989 026140 000001      .WORD      .S.      ;
9990 026140 000001      .WORD      .S.      ;
9991 026140 000001      .WORD      .S.      ;
9992 026140 000001      .WORD      .S.      ;
9993 026140 000001      .WORD      .S.      ;
9994 026140 000001      .WORD      .S.      ;
9995 026140 000001      .WORD      .S.      ;
9996 026140 000001      .WORD      .S.      ;
9997 026140 000001      .WORD      .S.      ;
9998 026140 000001      .WORD      .S.      ;
9999 026140 000001      .WORD      .S.      ;
10000 026140 000001      .WORD      .S.      ;

```

9916	026140	004000	.WORD .S	
9917	026142		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	016400	.WORD .S	
9944	026164		MOVE # 0, MEM MARINC	
9945		000013	MICPC=MICPC+1	
9946	026164	016400	.WORD .S	
9947	026166		MOVE # 0, MEM MARINC	;RESULT WITH C BIT SET.
9948		000014	MICPC=MICPC+1	
9949	026166	016400	.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	026174		MOVE # -1, MEM MARINC	
9957		000017	MICPC=MICPC+1	
9958	026174	016777	.WORD .S	
9959			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.	

DZXCA MACY11 27(1006) 13-MAY-77 14:07 PAGE 186
 DZXCA.P11 13-MAY-77 13:58 KMC11 ALU TESTS

PAGE 0208

```

10028 026254          MOVE      # 252, MEM MARINC      ;LOAD THE DATA IN MEMORY.
10029          MICPC=MICPC+1
10030 026254 000047   .WORD      .S.
10031 026256 016652   MOVE      # 252, MEM MARINC      ;RESULT WITH C BIT SET.
10032          MICPC=MICPC+1
10033 026256 000050   .WORD      .S.
10034 026260 016652   MOVE      # 7, SPAD <7>        ;SET THE COUNT.
10035          MICPC=MICPC+1
10036 026260 003007   .WORD      .S.
10037
10038 026262          MOVE      # 0, MLR              ;MAR+0.
10039          MICPC=MICPC+1
10040 026262 010000   .WORD      .S.
10041
10042 026264          25: MOVE      # 0, BREG              ;
10043          MICPC=MICPC+1
10044 026264 000053   .WORD      .S.
10045 026266 000400   SADD     SPAD <1>, BREG        ;CLEAR C BIT.
10046          MICPC=MICPC+1
10047 026266 060401   .WORD      .S.!.00
10048 026270          MOVE      MEM, SPAD <0> MARINC      ;GET THE FIRST OPERAND.
10049          MICPC=MICPC+1
10050 026270 057220   .WORD      .S.
10051 026272          OR      MEM, SPAD <J>, BR.SP, MARINC      ;SPAD <0>:=SFUNC.
10052          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          MICPC=MICPC+1
10059 026274 040360   .WORD      .S.
10060 026276          BZ      3S
10061          MICPC=MICPC+1
10062 026276 101473   .WORD      .S.
10063 026300          MOVE      MEM, OUT1 <4>        ;GOOD DATA
10064          MICPC=MICPC+1
10065 026300 041224   .WORD      .S.
10066 026302          MOVE      BREG, OUT1 <5>        ;BAD DATA.
10067          MICPC=MICPC+1
10068 026302 061225   .WORD      .S.
10069 026304          MOVE      # 15, BREG          ;SET TYPE OF ERROR.
10070          MICPC=MICPC+1
10071 026304 000415   .WORD      .S.
10072 026306          MOVE      BREG, OUT1 <3>        ;SET TYPE OF ERROR.
10073          MICPC=MICPC+1
10074 026306 061223   .WORD      .S.
10075 026310          MOVE      # 14, BREG          ;LOAD FUNCTION CODE...
10076          MICPC=MICPC+1
10077 026310 000414   .WORD      .S.
10078 026312          MOVE      BREG, OUT1 <CSR7>      ;LOAD IT...
10079          MICPC=MICPC+1
10080 026312 061227   .WORD      .S.
10081 026314          CALL     EROR              ;ALU A OR B ERROR...
10082 026314          MOVE      # <MICPC+3>, BREG
10083          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	0702C4	.WORD .S.	
10091	026322		SBR 2S	;LOOP ON ERROR...
10092		000072	MICPC=MICPC+1	
10093	026322	100453	.WORD .S.	
10094	026324		3S: 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 2S	;SCOPE THE DATA....
10105		000076	MICPC=MICPC+1	
10106	026332	100453	.WORD .S.	
10107	026334		6S: 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> 9S	;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 9S	
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          MICPC=MICPC+1
10142 026360 000423   .WORD      .S.
10143 026362          MOVE      BREG,OUT1 <CSR3>          ;
10144          MICPC=MICPC+1
10145 026362 061223   .WORD      .S.
10146 026364          MOVE      # 14,BREG          ;ALU FUNCTION CODE.
10147          MICPC=MICPC+1
10148 026364 000414   .WORD      .S.
10149 026366          MOVE      BREG,OUT1 <CSR7>          ;LOAD IT...
10150          MICPC=MICPC+1
10151 026366 061227   .WORD      .S.
10152 026370          CALL      EROR          ;REPORT ERROR...
10153 026370          MOVE      # <MICPC+3>,BREG
10154          MICPC=MICPC+1
10155 026370 000517   .WORD      .S.
10156 026372          SBR      EROR
10157          MICPC=MICPC+1
10158 026372 104400   .WORD      .S.
10159 026374          MOVE      SPAD <4>,MLR          ;RESTORE DATA POINTER.
10160          MICPC=MICPC+1
10161 026374 070204   .WORD      .S.
10162 026376          SBR      6S          ;LOOP ON ERROR...
10163          MICPC=MICPC+1
10164 026376 100477   .WORD      .S.
10165 026400 95:     CALL      SCP1          ;SCOPE THE ERROR...
10166 026400          MOVE      # <MICPC+3>,BREG
10167          MICPC=MICPC+1
10168 026400 000523   .WORD      .S.
10169 026402          SBR      SCP1
10170          MICPC=MICPC+1
10171 026402 104427   .WORD      .S.
10172 026404          MOVE      SPAD <4>,MLR          ;RESTORE DATA POINTER...
10173          MICPC=MICPC+1
10174 026404 070204   .WORD      .S.
10175 026406          SBR      6S          ;SCOPE THE DATA...
10176          MICPC=MICPC+1
10177 026406 100477   .WORD      .S.
10178 026410          MOVE      # 4,BREG          ;UPDATE BACKGROUND POINTER.
10179          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          MICPC=MICPC+1
10185 026414 063167   .WORD      .S!.DSP
10186 026416          BZ      4S          ;YES, SCOPE THE TEST.
10187          MICPC=MICPC+1
10188 026416 101532   .WORD      .S.
10189 026420          SBR      2S          ;M), THE NEXT.
10190          MICPC=MICPC+1
10191 026420 100453   .WORD      .S.
10192 026422 45:     CALL      SCOPE          ;SCOPE THE TEST...
10193 026422          MOVE      # <MICPC+3>,BREG
10194          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 IS ;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          STSTN
10217          ; TEST 47
10218
10219 026430 012737 000047 001202 TST47: MOV #47,STSTNM ; 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 KNC11 ADDRESS
10222 026444 004737 035536 JSR PC,LDVMT ;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          IS: MOVE #0,MLR ;SET MAR+LO.
10230          MICPC=MICPC+1
10231 026464 010000 .WORD .S.
10232 026466 MOVE #0,MPR ;SET MAR+HI.
10233          MICPC=MICPC+1
10234 026466 004000 .WORD .S.
10235 026470 MOVE #0,BREG ;
10236          MICPC=MICPC+1
10237 026472 .WORD .S.
10238          MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS...
10239          MICPC=MICPC+1
10240 026474 .WORD .S.
10241          MOVE BREG,SPAD <0> ;
10242          MICPC=MICPC+1
10243 026476 .WORD .S.
10244          MOVE BREG,SPAD <1> ;
10245          MICPC=MICPC+1
10246 026500 .WORD .S.
10247          MOVE BREG,SPAD <2> ;
10248          MICPC=MICPC+1
10249 026502 .WORD .S.
10250          SDEC SPAD <2> ;
10251          MICPC=MICPC+1
          .WORD .S!.DSP

```

10252	026504	000010	MOVE BREG SPAD <4>	;
10253		063224	MICPC=MICPC+1	
10254	026504		.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> 3S ;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 3S
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 2S ;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

35:

10420	026656	070204	.WORD .S.	
10421	026660		SBR 25	; 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          MOVE      SPAD (4),MLR          ;RESTORE DATA POINTER.
10477 000117          MICPC=MICPC+1
10478 026722 070204    .WORD      .S.
10479 026724          SBR        65          ;LOOP ON ERROR...
10480          MICPC=MICPC+.
10481 026724 100477    .WORD      .S.
10482 026726          95:      CALL      SCP1          ;SCOPE THE ERROR...
10483 026726          MOVE      # <MICPC+3>,BREG
10484          MICPC=MICPC+1
10485 026726 000523    .WORD      .S.
10486 026730          SBR        SCP1
10487          MICPC=MICPC+1
10488          .WORD      .S.
10489 026730 104427    MOVE      SPAD (4),MLR          ;RESTORE DATA POINTER...
10490          MICPC=MICPC+1
10491 026732 070204    .WORD      .S.
10492 026734          SBR        65          ;SCOPE THE DATA...
10493          MICPC=MICPC+1
10494 026734 100477    .WORD      .S.
10495 026736          MOVE      # 4,BREG          ;UPDATE BACKGROUND POINTER.
10496          MICPC=MICPC+1
10497 026736 000404    .WORD      .S.
10498 026740          SADD      BREG,SPAD (4),MARINC ;ALSO DATA POINTER.
10499 026740 077004    .WORD      .S,MARINC!.DO
10500 026742          SDEC      SPAD (7)          ;IS IT DONE??
10501          MICPC=MICPC+1
10502 026742 063167    .WORD      .S!.DSP
10503 026744          BZ          45          ;YES, SCOPE THE TEST.
10504          MICPC=MICPC+1
10505 026744 101532    .WORD      .S.
10506 026746          SBR        25          ;DO, THE NEXT.
10507          MICPC=MICPC+1
10508 026746 100453    .WORD      .S.
10509 026750          45:      CALL      SCOPE          ;SCOPE THE TEST...
10510 026750          MOVE      # <MICPC+3>,BREG
10511          MICPC=MICPC+1
10512 026750 000533    .WORD      .S.
10513 026752          SBR        SCOPE
10514          MICPC=MICPC+1
10515 026752 104454    .WORD      .S.
10516 026754          SBR        15          ;DO THE NEXT ITERATION...
10517          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          ;***** TEST 50 *****
10524          ;*ALU TEST
10525          ;*TEST OF ALU FUNCTION ADD WITH C BIT CLEARED.
10526          ;*TEST OF ALU FUNCTION ADD WITH C BIT SET.
10527          ;*ALU FUNCTION (A PLUS B)
10528          ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
10529          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
10530 026756          SXZ
10531          ;*****

```

```

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
10539 026772 004737 035536 JSR PC,LDRWT        ;R1 CONTAINS BASE KMC11 ADDRESS
10540 026776 027012 MCT50              ;LOAD-VERIFY-WAIT.
10541 027000 104022 ERROR              ;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 1S: MOVE #0,MLR          ;SET MAR+LO.
10546 000000 MICPC=MICPC+1
10547 027012 010000 .WORD .S.
10548 027014 MOVE #0,MPR          ;SET MAR+HI.
10549 000001 MICPC=MICPC+1
10550 027014 004000 .WORD .S.
10551 027016 MOVE #0,BREG
10552 000002 MICPC=MICPC+1
10553 027016 000400 .WORD .S.
10554 027020 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS...
10555 000003 MICPC=MICPC+1
10556 027020 063236 .WORD .S.
10557 027022 MOVE BREG,SPAD <0> ;
10558 000004 MICPC=MICPC+1
10559 027022 063220 .WORD .S.
10560 027024 MOVE BREG,SPAD <1> ;
10561 000005 MICPC=MICPC+1
10562 027024 063221 .WORD .S.
10563 027026 MOVE BREG,SPAD <2> ;
10564 000006 MICPC=MICPC+1
10565 027026 063222 .WORD .S.
10566 027030 SOEC SPAD <2> ;
10567 000007 MICPC=MICPC+1
10568 027030 063162 .WORD .S. DSP
10569 027032 MOVE BREG,SPAD <4> ;
10570 000010 MICPC=MICPC+1
10571 027032 063224 .WORD .S.
10572 027034 MOVE #0,MEM MARINC ;LOAD THE DATA IN MEMORY.
10573 000011 MICPC=MICPC+1
10574 027034 016400 .WORD .S.
10575 027036 MOVE #0,MEM MARINC ;LOAD THE DATA IN MEMORY
10576 000012 MICPC=MICPC+1
10577 027036 016400 .WORD .S.
10578 027040 MOVE #0,MEM MARINC
10579 000013 MICPC=MICPC+1
10580 027040 016400 .WORD .S.
10581 027042 MOVE #0,MEM MARINC ;RESULT WITH C BIT SET.
10582 000014 MICPC=MICPC+1
10583 027042 016400 .WORD .S.
10584 027044 MOVE #-1,MEM MARINC ;LOAD THE DATA IN MEMORY.
10585 000015 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	000041	MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
10645		016525	MICPC=MICPC+1 .WORD .S.
10646	027114		
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	016574	.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. DD
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 000104 MICPC=MICPC+1
10758 027222 .WORD $.
10759 027224 SIFEQ MEM,SPAD <0> 95 ;BR IF GOOD...
10760
10761
10762 027224 SUB2C SPAD <0>,MEM,NOP
10763 000105 MICPC=MICPC+1
10764 027224 .WORD $.
10765 027226 BZ 95
10766 000106 MICPC=MICPC+1
10767 027226 101521 .WORD $.
10768 027230 MOVE MEM,OUT1 <CSR4> ;GOOD DATA.
10769 000107 MICPC=MICPC+1
10770 027230 .WORD $.
10771 027232 MOVE BREG,OUT1 <CSR5> ;BAD DATA.
10772 000110 MICPC=MICPC+1
10773 027232 .WORD $.
10774 027234 MOVE # 23,BREG ;ERROR TYPE...
10775 000111 MICPC=MICPC+1
10776 027234 000423 .WORD $.
10777 027236 MOVE BREG,OUT1 <CSR3> ;
10778 000112 MICPC=MICPC+1
10779 027236 061223 .WORD $.
10780 027240 MOVE # 00,BREG ;ALU FUNCTION CODE.
10781 000113 MICPC=MICPC+1
10782 027240 000400 .WORD $.
10783 027242 MOVE BREG,OUT1 <CSR7> ;LOAD IT...
10784 000114 MICPC=MICPC+1
10785 027242 061227 .WORD $.
10786 027244 CALL EROR ;REPORT ERROR...
10787 027244 MOVE # <MICPC+3>,BREG
10788 000115 MICPC=MICPC+1
10789 027244 000517 .WORD $.
10790 027246 SBR EROR
10791 000116 MICPC=MICPC+1
10792 027246 104400 .WORD $.
10793 027250 MOVE SPAD <4>,MLR ;RESTORE DATA POINTER.
10794 000117 MICPC=MICPC+1
10795 027250 070204 .WORD $.
10796 027252 SBR 65 ;LOOP ON ERROR...
10797 000120 MICPC=MICPC+1
10798 027252 100477 .WORD $.
10799 027254 95: CALL SCP1 ;SCOPE THE ERROR...
10800 027254 MOVE # <MICPC+3>,BREG
10801 000121 MICPC=MICPC+1
10802 027254 000523 .WORD $.
10803 027256 SBR SCP1
10804 000122 MICPC=MICPC+1
10805 027256 104427 .WORD $.
10806 027260 MOVE SPAD <4>,MLR ;RESTORE DATA POINTER...
10807 000123 MICPC=MICPC+1
10808 027260 070204 .WORD $.
10809 027262 SBR 65 ;SCOPE THE DATA...
10810 000124 MICPC=MICPC+1
10811 027262 100477 .WORD $.
    
```

```

10812 027264 MOVE # 4,BREG ;UPDATE BACKGROUND POINTER.
10813 000125 MICPC=MICPC+1
10814 027264 .WORD .S.
10815 027266 SADD BREG,SPAD (4),MARINC ;ALSO DATA POINTER.
10816 027266 .WORD .S. !MARINC!.DO
10817 027270 SDEC SPAD (7) ;IS IT DONE??
10818 000126 MICPC=MICPC+1
10819 027270 .WORD .S.!.DSP
10820 027272 BZ 45 ;YES, SCOPE THE TEST.
10821 000127 MICPC=MICPC+1
10822 027272 .WORD .S.
10823 027274 SBR 23 ;DO, THE NEXT.
10824 000130 MICPC=MICPC+1
10825 027274 .WORD .S.
10826 027276 45: CALL SCPE ;SCOPE THE TEST...
10827 027276 MOVE # (MICPC+3),BREG
10828 000131 MICPC=MICPC+1
10829 027276 .WORD .S.
10830 027300 SBR SCPE
10831 000132 MICPC=MICPC+1
10832 027300 .WORD .S.
10833 027302 SBR 15 ;DO THE NEXT ITERATION...
10834 000133 MICPC=MICPC+1
10835 027302 .WORD .S.
10836 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
10837 027304 SXZ
10838
10839
10840 ;***** TEST 51 *****
10841 ;ALU TEST
10842 ;TEST OF ALU FUNCTION 2A W/C WITH C BIT CLEARED.
10843 ;TEST OF ALU FUNCTION 2A W/C WITH C BIT SET.
10844 ;ALU FUNCTION (A PLUS A PLUS C)
10845 ;LOAD MAIN MEMORY 16 WORDS OF DATA.
10846 ;PERFORM THE FUNCTION, VERIFY THE RESULTS..
10847 027304 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 #ST52,NEXT ; POINT TO THE START OF NEXT TEST.
10855 ;R1 CONTAINS BASE KMC11 ADDRESS
10856 027320 004737 035536 JSR PC,LDRWRT ;LOAD-VERIFY-WAIT.
10857 027324 027340 MCT51
10858 027326 104022 ERROR 22 ;TIME OUT ERROR...
10859 027330 012706 001200 MOV #STACK,SP ;RESET STACK...
10860 027334 000177 152102 JMP @NEXT ;GO TO NEXT TEST...
10861 027340 MCT51:
10862 027340 IS: MOVE # 0,MLR ;SET MAR+LO.
10863 000000 MICPC=MICPC+1
10864 027340 010000 .WORD .S.
10865 027342 MOVE # 0,MPR ;SET MAR+HI.
10866 000001 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

DZKCA MACY11 27(1006) 13-MAY-77 14:07 PAGE 202
 DZKCA.P11 13-MAY-77 13:58 KMC11 ALU TESTS

PAGE

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	027466	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		SIFEQ 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 <CSR7> ;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          MICPC=MICPC+1
11038 027520 104400  .WORD   $
11039 027522          MOVE    SPAD <4>,MLR      ;RESET DATA POINTER...
11040          MICPC=MICPC+1
11041 027522 070204  .WORD   $
11042 027524          SBR      25          ;LOOP ON ERROR...
11043          MICPC=MICPC+1
11044 027524 100453  .WORD   $
11045 027526          CALL    SCP1
11046 027526          MOVE    # <MICPC+3>,BREG
11047          MICPC=MICPC+1
11048 027526 000475  .WORD   $
11049 027530          SBR      SCP1
11050          MICPC=MICPC+1
11051 027530 000074  .WORD   $
11052 027532          MOVE    SPAD <4>,MLR      ;
11053          MICPC=MICPC+1
11054 027532 070204  .WORD   $
11055 027534          SBR      25          ;SCOPE THE DATA....
11056          MICPC=MICPC+1
11057 027534 100453  .WORD   $
11058 027536          MOVE    SPAD <4>,MLR      ;RESET DATA POINTER...
11059          MICPC=MICPC+1
11060 027536 070204  .WORD   $
11061 027540          MOVE    MEM,SPAD <0>,MARINC    ;GET FIRST OPRAND...
11062          MICPC=MICPC+1
11063 027540 057220  .WORD   $
11064 027542          MOVE    # 377,BREG      ;
11065          MICPC=MICPC+1
11066 027542 000777  .WORD   $
11067 027544          SADD    SPAD <2>,BREG    ;SET C BIT...
11068          MICPC=MICPC+1
11069 027544 060402  .WORD   $;!..DO
11070 027546          SROL    SPAD <0>,BR.SP,MARINC    ;SP 0 & BR = 2A W/C
11071          MICPC=MICPC+1
11072 027546 077540  .WORD   $
11073 027550          MOVE    MEM,SPAD <3>,MARINC    ;DUMMY INSTR, TO MARINC.
11074          MICPC=MICPC+1
11075 027550 057223  .WORD   $
11076 027552          SIFEQ  MEM,SPAD <0> 9S      ;BR IF GOOD...
11077
11078
11079 027552          SUB2C  SPAD <0>,MEM,NOP
11080          MICPC=MICPC+1
11081 027552 040360  .WORD   $
11082 027554          BZ      9S
11083          MICPC=MICPC+1
11084 027554 101521  .WORD   $
11085 027556          MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
11086          MICPC=MICPC+1
11087 027556 041224  .WORD   $
11088 027560          MOVE    BREG,OUT1 <CSR5>    ;BAD DATA.
11089          MICPC=MICPC+1
11090 027560 061225  .WORD   $
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.
95: 11116 027602 CALL SCPE ;SCOPE THE ERROR...
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!.D0
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.
45: 11143 027624 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          ;***** TEST 52 *****
11158          ;*ALU TEST
11159          ;*TEST OF ALU FUNCTION SUB WITH C BIT CLEARED.
11160          ;*TEST OF ALU FUNCTION SUB WITH C BIT SET.
11161          ;*ALU FUNCTION (A-B)
11162          ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
11163          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
11164 027632          SXZ
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,LDVWRT          ;LOAD-VERIFY-WAIT.
11174 027652 027666 MCT52
11175 027654 104022 ERROR
11176 027656 012706 001200 MOV #STACK,SP          ;TIME OUT ERROR...
11177 027662 000177 151554 JMP @NEXT          ;RESET STACK...
11178 027666 MCT52:          ;GO TO NEXT TEST...
11179 027666 15: 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 000005          .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          MICPC=MICPC+1
11262 027754 016400  .WORD .S.
11263 027756          MOVE # 0, MEM MARINC ;RESULT WITH C BIT SET.
11264          MICPC=MICPC+1
11265 027756 016400  .WORD .S.
11266 027760          MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
11267          MICPC=MICPC+1
11268 027760 016652  .WORD .S.
11269 027762          MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
11270          MICPC=MICPC+1
11271 027762 016525  .WORD .S.
11272 027764          MOVE # 125, MEM MARINC
11273          MICPC=MICPC+1
11274 027764 016525  .WORD .S.
11275 027766          MOVE # 125, MEM MARINC ;RESULT WITH C BIT SET.
11276          MICPC=MICPC+1
11277 027766 016525  .WORD .S.
11278 027770          MOVE # 125, MEM MARINC ;LOAD THE DATA IN MEMORY.
11279          MICPC=MICPC+1
11280 027770 016525  .WORD .S.
11281 027772          MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY
11282          MICPC=MICPC+1
11283 027772 016652  .WORD .S.
11284 027774          MOVE # 253, MEM MARINC
11285          MICPC=MICPC+1
11286 027774 016653  .WORD .S.
11287 027776          MOVE # 253, MEM MARINC ;RESULT WITH C BIT SET.
11288          MICPC=MICPC+1
11289 027776 016653  .WORD .S.
11290 030000          MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
11291          MICPC=MICPC+1
11292 030000 016652  .WORD .S.
11293 030002          MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
11294          MICPC=MICPC+1
11295 030002 016652  .WORD .S.
11296 030004          MOVE # 0, MEM MARINC ;LOAD THE DATA IN MEMORY.
11297          MICPC=MICPC+1
11298 030004 016400  .WORD .S.
11299 030006          MOVE # 0, MEM MARINC ;RESULT WITH C BIT SET.
11300          MICPC=MICPC+1
11301 030006 016400  .WORD .S.
11302 030010          MOVE # 7, SPAD <7> ;SET THE COUNT.
11303          MICPC=MICPC+1
11304 030010 003007  .WORD .S.
11305
11306 030012          MOVE # 0, MLR ;MAR+0.
11307          MICPC=MICPC+1
11308 030012 010000  .WORD .S.
11309
11310 030014          MOVE # 0, BREG ;
11311          MICPC=MICPC+1
11312 030014 000400  .WORD .S.
11313 030016          $ADD SPAD <1>, BREG ;CLEAR C BIT.
11314          MICPC=MICPC+1
11315 030016 060401  .WORD .S.!.DD
    
```

DZKCA MACY11 27(1006) 13-MAY-77 14:07 PAGE 209
 DZKCA.P11 13-MAY-77 13:58 KMC11 ALU TESTS

```

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>:=$FUNC.
11320 000056 MICPC=MICPC+1
11321 030022 057740 .WORD .S.
11322 030024 SIFEQ MEM,SPAD <0> 3S ;BRANCH IF GOOD.
11323 000057
11324 030024 SUBIC SPAD <0>,MEM,NOP
11325 000057 MICPC=MICPC+1
11326 030024 040360 .WORD .S.
11327 030024 040360 BZ 3S
11328 030026 MICPC=MICPC+1
11329 000060 .WORD .S.
11330 030026 101473 MOVE MEM,OUT1 <4> ;GOOD DATA
11331 030030 MICPC=MICPC+1
11332 000061 .WORD .S.
11333 030030 041224 MOVE BREG,OUT1 <5> ;BAD DATA.
11334 030032 MICPC=MICPC+1
11335 000062 .WORD .S.
11336 030032 061225 MOVE # 15,BREG ;SET TYPE OF ERROR.
11337 030034 MICPC=MICPC+1
11338 000063 .WORD .S.
11339 030034 000415 MOVE BREG,OUT1 <3> ;SET TYPE OF ERROR.
11340 030036 MICPC=MICPC+1
11341 000064 .WORD .S.
11342 030036 061223 MOVE # 16,BREG ;LOAD FUNCTION CODE...
11343 030040 MICPC=MICPC+1
11344 000065 .WORD .S.
11345 030040 000416 MOVE BREG,OUT1 <CSR7> ;LOAD IT...
11346 030042 MICPC=MICPC+1
11347 000066 .WORD .S.
11348 030042 061227 CALL EROR ;ALU SUB ERROR...
11349 030044 MOVE # <MICPC+3>,BREG
11350 030044 MICPC=MICPC+1
11351 000067 .WORD .S.
11352 000444 SBR EROR
11353 000446 MICPC=MICPC+1
11354 000446 .WORD .S.
11355 000446 MOVE SPAD <4>,MLR ;RESET DATA POINTER...
11356 000446 MICPC=MICPC+1
11357 000446 .WORD .S.
11358 000446 SBR 2S ;LOOP ON ERROR...
11359 000446 MICPC=MICPC+1
11360 000072 .WORD .S.
11361 000453 CALL SCP1
11362 030054 MOVE # <MICPC+3>,BREG
11363 000073 MICPC=MICPC+1
11364 030054 000475 .WORD .S.
11365 030056 SBR SCP1
11366 000074 MICPC=MICPC+1
11367 000475 .WORD .S.
11368 030056 104427 MOVE SPAD <4>,MLR ;
11369 030060 MICPC=MICPC+1
11370 000075 .WORD .S.
11371 030060 070204
    
```

3S:

DZXCA MACY11 27(1006) 13-MAY-77 14:07 PAGE 210
 DZXCA.P11 13-MAY-77 13:58 KMC11 ALU TESTS

```

11372 030062          SBR          28          ; SCOPE THE DATA....
11373 030062 000076  MICPC=MICPC+1
11374 030062 100453  .WORD .S.
11375 030064          6S: MOVE SPAD <4>,MLR          ;RESET DATA POINTER...
11376 030064 000077  MICPC=MICPC+1
11377 030064 070204  .WORD .S.
11378 030066          MOVE MEM,SPAD <0>,MARINC          ;GET FIRST OPRAND...
11379 030066 000100  MICPC=MICPC+1
11380 030066 057220  .WORD .S.
11381 030070          MOVE # 377,BREG          ;
11382 030070 000101  MICPC=MICPC+1
11383 030070 000777  .WORD .S.
11384 030072          SPAD <2>,BREG          ;SET C BIT...
11385 030072 000102  MICPC=MICPC+1
11386 030072 060402  .WORD .S.! .DO
11387 030074          SUBC MEM,SPAD <0>,BR.SP,MARINC          ;SP 0 & BR = SUB
11388 030074 000103  MICPC=MICPC+1
11389 030074 057740  .WORD .S.
11390 030076          MOVE MEM,SPAD <3>,MARINC          ;DUMMY INSTR, TO MARINC.
11391 030076 000104  MICPC=MICPC+1
11392 030076 057223  .WORD .S.
11393 030100          SIFEQ MEM,SPAD <0>          9S          ;9R IF GOOD...
11394
11395
11396 030100          SUBC SPAD <0>,MEM,NOP
11397 030100 000105  MICPC=MICPC+1
11398 030100 040360  .WORD .S.
11399 030102          BZ          9S
11400 030102 000106  MICPC=MICPC+1
11401 030102 101521  .WORD .S.
11402 030104          MOVE MEM,OUT1 <CSR4>          ;GOOD DATA.
11403 030104 000107  MICPC=MICPC+1
11404 030104 041224  .WORD .S.
11405 030106          MOVE BREG,OUT1 <CSR5>          ;BAD DATA.
11406 030106 000110  MICPC=MICPC+1
11407 030106 061225  .WORD .S.
11408 030110          MOVE # 23,BREG          ;ERROR TYPE...
11409 030110 000111  MICPC=MICPC+1
11410 030110 000423  .WORD .S.
11411 030112          MOVE BREG,OUT1 <CSR3>          ;
11412 030112 000112  MICPC=MICPC+1
11413 030112 061223  .WORD .S.
11414 030114          MOVE # 16,BREG          ;ALU FUNCTION CODE.
11415 030114 000113  MICPC=MICPC+1
11416 030114 000416  .WORD .S.
11417 030116          MOVE BREG,OUT1 <CSR7>          ;LOAD IT...
11418 030116 000114  MICPC=MICPC+1
11419 030116 061227  .WORD .S.
11420 030120          CALL EROR          ;REPORT ERROR...
11421 030120          MOVE # <MICPC+3>,BREG
11422 030120 000115  MICPC=MICPC+1
11423 030120 000517  .WORD .S.
11424 030122          SBR EROR
11425 030122 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: CALL SCP1 ;SCOPE THE ERROR...
11435 000121
11436 030130 000523
11437 030132
11438 000122
11439 030132 104427
11440 030134
11441 000123
11442 030134 070204
11443 030136
11444 000124
11445 030136 100477
11446 030140
11447 000125
11448 030140 000404
11449 030142
11450 030142 077004
11451 030144
11452 000126
11453 030144 063167 .WORD .S!.DSP
11454 030146 BZ 45 ;YES, SCOPE THE TEST.
11455 000127
11456 030146 101532
11457 030150 SBR 25 ;DO, THE NEXT.
11458 000130
11459 030150 100453
11460 030152 45: CALL SCOPE ;SCOPE THE TEST...
11461 030152
11462 000131
11463 030152 000533
11464 030154
11465 000132
11466 030154 104454
11467 030156
11468 000133
11469 030156 100400
11470 030160 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
11471 030160 SXZ
11472
11473
11474
11475
11476
11477
11478
11479
11480
11481 030160 SXZ
11482
11483

```

MICPC=MICPC+1
 .WORD .S.
 SBR 65 ;LOOP ON ERROR...
 MICPC=MICPC+1
 .WORD .S.
 CALL SCP1 ;SCOPE THE ERROR...
 MOVE # <MICPC+3>, BREG
 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 65 ;SCOPE THE DATA...
 MICPC=MICPC+1
 .WORD .S.
 MOVE # 4, BREG ;UPDATE BACKGROUND POINTER.
 MICPC=MICPC+1
 .WORD .S.
 \$ADD BREG, SPAD <4>, MARINC ;ALSO DATA POINTER.
 .WORD .S. !MARINC!.DO
 \$DEC SPAD <7> ;IS IT DONE??
 MICPC=MICPC+1
 .S!.DSP
 BZ 45 ;YES, SCOPE THE TEST.
 MICPC=MICPC+1
 .WORD .S.
 SBR 25 ;DO, THE NEXT.
 MICPC=MICPC+1
 .WORD .S.
 CALL SCOPE ;SCOPE THE TEST...
 MOVE # <MICPC+3>, BREG
 MICPC=MICPC+1
 .WORD .S.
 SBR SCOPE
 MICPC=MICPC+1
 .WORD .S.
 SBR 15 ;DO THE NEXT ITERATION...
 .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..
 ;:*****

```

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     @ST54,NEXT        ; POINT TO THE START OF NEXT TEST.
11489          ;R1 CONTAINS BASE KMC11 ADDRESS
11490 030174 004737 035536          JSR     PC,LDRMT         ;LOAD-VERIFY-WAIT.
11491 030200 030214          MCT53
11492 030202 104022          ERROR
11493 030204 012706 001200          MOV     @STACK,SP       ;TIME OUT ERROR...
11494 030210 000177 151226          JMP     @NEXT           ;RESET STACK.
11495 030214          MCT53:                ;GO TO NEXT TEST...
11496 030214          IS:          MOVE     # 0,MLR          ;SET MAR+LO.
11497          000000          NICPC=NICPC+1
11498 030214 010000          .WORD  .S.
11499 030216          MOVE     # 0,MPR          ;SET MAR+HI.
11500          000001          NICPC=NICPC+1
11501 030216 004000          .WORD  .S.
11502 030220          MOVE     # 0,BREG
11503          000002          NICPC=NICPC+1
11504 030220 000400          .WORD  .S.
11505 030222          MOVE     @BREG,SPAD (16) ;FOR RETURN ADDRESS...
11506          000003          NICPC=NICPC+1
11507 030222 063236          .WORD  .S.
11508 030224          MOVE     @BREG,SPAD (0) ;
11509          000004          NICPC=NICPC+1
11510 030224 063220          .WORD  .S.
11511 030226          MOVE     @BREG,SPAD (1) ;
11512          000005          NICPC=NICPC+1
11513 030226 063221          .WORD  .S.
11514 030230          MOVE     @BREG,SPAD (2) ;
11515          000006          NICPC=NICPC+1
11516 030230 063222          .WORD  .S.
11517 030232          @DEC SPAD (2)
11518          000007          NICPC=NICPC+1
11519 030232 063162          .WORD  .S.!.DSP
11520 030234          MOVE     @BREG,SPAD (4) ;
11521          000010          NICPC=NICPC+1
11522 030234 063224          .WORD  .S.
11523 030236          MOVE     # 0, MEM MARINC ;LOAD THE DATA IN MEMORY.
11524          000011          NICPC=NICPC+1
11525 030236 016400          .WORD  .S.
11526 030240          MOVE     # 0, MEM MARINC ;LOAD THE DATA IN MEMORY
11527          000012          NICPC=NICPC+1
11528 030240 016400          .WORD  .S.
11529 030242          MOVE     # 0, MEM MARINC
11530          000013          NICPC=NICPC+1
11531 030242 016400          .WORD  .S.
11532 030244          MOVE     # 1, MEM MARINC ;RESULT WITH C BIT SET.
11533          000014          NICPC=NICPC+1
11534 030244 016401          .WORD  .S.
11535 030246          MOVE     # -1, MEM MARINC ;LOAD THE DATA IN MEMORY.
11536          000015          NICPC=NICPC+1
11537 030246 016777          .WORD  .S.
11538 030250          MOVE     # 0, MEM MARINC ;LOAD THE DATA IN MEMORY.
11539          000016          NICPC=NICPC+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		000041	MICPC=MICPC+1	
11597	030316	016525	.WORD .S.	
11598	030320		MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY
11599		000042	MICPC=MICPC+1	
11600	030320	016652	.WORD .S.	
11601	030322		MOVE # -1, MEM MARINC	
11602		000043	MICPC=MICPC+1	
11603	030322	016777	.WORD .S.	
11604	030324		MOVE # 0, MEM MARINC	;RESULT WITH C BIT SET.
11605		000044	MICPC=MICPC+1	
11606	030324	016400	.WORD .S.	
11607	030326		MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY.
11608		000045	MICPC=MICPC+1	
11609	030326	016652	.WORD .S.	
11610	030330		MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY.
11611		000046	MICPC=MICPC+1	
11612	030330	016652	.WORD .S.	
11613	030332		MOVE # 124, MEM MARINC	;LOAD THE DATA IN MEMORY.
11614		000047	MICPC=MICPC+1	
11615	030332	016524	.WORD .S.	
11616	030334		MOVE # 125, MEM MARINC	;RESULT WITH C BIT SET.
11617		000050	MICPC=MICPC+1	
11618	030334	016525	.WORD .S.	
11619	030336		MOVE # 7, SPAD <7>	;SET THE COUNT.
11620		000051	MICPC=MICPC+1	
11621	030336	003007	.WORD .S.	
11622				
11623	030340		MOVE # 0, MLR	;MAR←0.
11624		000052	MICPC=MICPC+1	
11625	030340	010000	.WORD .S.	
11626				
11627	030342		25: MOVE # 0, BREG	;
11628		000053	MICPC=MICPC+1	
11629	030342	000400	.WORD .S.	
11630	030344		SADD SPAD <1>, BREG	;CLEAR C BIT.
11631		000054	MICPC=MICPC+1	
11632	030344	060401	.WORD .S.!.DO	
11633	030346		MOVE MEM, SPAD <0> MARINC	;GET THE FIRST OPERAND.
11634		000055	MICPC=MICPC+1	
11635	030346	057220	.WORD .S.	
11636	030350		ADDWC MEM, SPAD <0>, BR.SP, MARINC	;SPAD <0>:=SFUNC.
11637		000056	MICPC=MICPC+1	
11638	030350	057420	.WORD .S.	
11639	030352		\$IFEQ MEM, SPAD <0> 3S	;BRANCH IF GOOD.
11640				
11641				
11642	030352		SUB2C SPAD <0>, MEM, NOP	
11643		000057	MICPC=MICPC+1	
11644	030352	040360	.WORD .S.	
11645	030354		BZ 3S	
11646		000060	MICPC=MICPC+1	
11647	030354	101473	.WORD .S.	
11648	030356		MOVE MEM, OUT1 <4>	;GOOD DATA
11649		000061	MICPC=MICPC+1	
11650	030356	041224	.WORD .S.	
11651	030360		MOVE BREG, OUT1 <5>	;BAD DATA.

```

11652      000062      MICPC=MICPC+1
11653      030360      .WORD      $
11654      030362      MOVE      # 15, BREG      ;SET TYPE OF ERROR.
11655      000063      MICPC=MICPC+1
11656      030362      .WORD      $
11657      030364      MOVE      BREG, OUT1 <3> ;SET TYPE OF ERROR.
11658      000064      MICPC=MICPC+1
11659      030364      .WORD      $
11660      030366      MOVE      # 01, BREG      ;LOAD FUNCTION CODE...
11661      000065      MICPC=MICPC+1
11662      030366      .WORD      $
11663      03037 ]      MOVE      BREG, OUT1 <CSR7> ;LOAD IT...
11664      000066      MICPC=MICPC+1
11665      030370      .WORD      $
11666      030372      CALL      EROR      ;ALU ADD W/C ERROR...
11667      030372      MOVE      # <MICPC+3>, BREG
11668      000067      MICPC=MICPC+1
11669      030372      .WORD      $
11670      030374      SBR      EROR
11671      000070      MICPC=MICPC+1
11672      030374      .WORD      $
11673      030376      MOVE      SPAD <4>, MLR      ;RESET DATA POINTER...
11674      000071      MICPC=MICPC+1
11675      030376      .WORD      $
11676      030400      SBR      25      ;LOOP ON ERROR...
11677      000072      MICPC=MICPC+1
11678      030400      .WORD      $
11679      030402      35:      CALL      SCP1
11680      030402      MOVE      # <MICPC+3>, BREG
11681      000073      MICPC=MICPC+1
11682      030402      .WORD      $
11683      030404      SBR      SCP1
11684      000074      MICPC=MICPC+1
11685      030404      .WORD      $
11686      030406      MOVE      SPAD <4>, MLR      ;
11687      000075      MICPC=MICPC+1
11688      030406      .WORD      $
11689      030410      SBR      25      ; SCOPE THE DATA....
11690      000076      MICPC=MICPC+1
11691      030410      .WORD      $
11692      030412      65:      MOVE      SPAD <4>, MLR      ;RESET DATA POINTER...
11693      000077      MICPC=MICPC+1
11694      030412      .WORD      $
11695      030414      MOVE      MEM, SPAD <0>, MARINC      ;GET FIRST OPRAND...
11696      000100      MICPC=MICPC+1
11697      030414      .WORD      $
11698      030416      MOVE      # 377, BREG      ;
11699      000101      MICPC=MICPC+1
11700      030416      .WORD      $
11701      030420      SADD     SPAD <2>, BREG      ;SET C BIT...
11702      000102      MICPC=MICPC+1
11703      030420      .WORD      $, !, DO
11704      030422      ADDMC   MEM, SPAD <0>, BR.SP, MARINC      ;SP 0 & BR = ADD W/C
11705      000103      MICPC=MICPC+1
11706      030422      .WORD      $
11707      030424      MOVE     MEM, SPAD <3>, MARINC      ;DUMMY INSTR, TO MARINC.

```

```

11708      000104      MICPC=MICPC+1
11709 030424 057223      .WORD      .S.
11710 030426      $IFEQ      MEM,SPAD <0>      9$      ;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        9$
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 051227      .WORD      .S.
11737 030446      CALL      ERROR ;REPORT ERROR...
11738 030446      MOVE      # <MICPC+3>,BREG
11739      000115      MICPC=MICPC+1
11740 030446 000517      .WORD      .S.
11741 030450      SBR      ERROR
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      6$      ;LOOP ON ERROR...
11748      000120      MICPC=MICPC+1
11749 030454 100477      .WORD      .S.
11750 030456      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      6$      ;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      .WORD .S.
11766      030470      $ADD BREG,SPAD (<4>),MARINC ;ALSO DATA POINTER.
11767      030470      .WORD .S.!MARINC!.DO
11768      030472      $DEC SPAD (<7> ;IS IT DONE??
11769      000126      MICPC=MICPC+1
11770      030472      .WORD .S!.DSP
11771      030474      BZ 45 ;YES, SCOPE THE TEST.
11772      000127      MICPC=MICPC+1
11773      030474      .WORD .S.
11774      030476      $BR 25 ;DO, THE NEXT.
11775      000130      MICPC=MICPC+1
11776      030476      .WORD .S.
11777      030500      45: CALL SCOPE ;SCOPE THE TEST...
11778      030500      MOVE # (<MICPC+3>),BREG
11779      000131      MICPC=MICPC+1
11780      030500      .WORD .S.
11781      030502      $BR SCOPE
11782      000132      MICPC=MICPC+1
11783      030502      .WORD .S.
11784      030504      $BR 15 ;DO THE NEXT ITERATION...
11785      000133      MICPC=MICPC+1
11786      030504      .WORD .S.
11787      030506      SALUT1 0,<SUB W/C>,SUBWC,-1,0,376,377,0,1,-1,0,-1,0,124,125,252,253,-1,0,<A-B-C>,1,02
11788      030506      $XZ
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      $XZ
11799      ;*****K*****
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,LDRWMT ;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 IS: MOVE #0,MLR ; SET MAR+LO.
11814      000000 MICPC=MICPC+1
11815      030542 010000 .WORD .S.
11816      030544 MOVE #0,MPR ; SET MAR+HI.
11817      000001 MICPC=MICPC+1
11818      030544 004000 .WORD .S.
11819      030546 MOVE #0,BREG ;

```

11820		000002	MICPC=MICPC+1	
11821	030546	000400	.WORD .S	
11822	030550		MOVE #0, SPAD <16>	;FOR RETURN ADDRESS...
11823		000003	MICPC=MICPC+1	
11824	030550	063236	.WORD .S	
11825	030552		MOVE #0, SPAD <0>	;
11826		000004	MICPC=MICPC+1	
11827	030552	063220	.WORD .S	
11828	030554		MOVE #0, SPAD <1>	;
11829		000005	MICPC=MICPC+1	
11830	030554	063221	.WORD .S	
11831	030556		MOVE #0, 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 #0, 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		

11938	030660	016777	.WORD .S
11939	030662		MOVE # 0, MEM, MARINC ; RESULT WITH C BIT SET.
11940		000050	MICPC=MICPC+1
11941	030662	016400	.WORD .S
11942	030664		MOVE # 7, SPAD <7> ; SET THE COUNT.
11943		000051	MICPC=MICPC+1
11944	030664	003007	.WORD .S
11945			
11946	030666		MOVE # 0, MLR ; MAR+0.
11947		000052	MICPC=MICPC+1
11948	030666	010000	.WORD .S
11949			
11950	030670		2S: MOVE # 0, BREG ;
11951		000053	MICPC=MICPC+1
11952	030670	000400	.WORD .S
11953	030672		SADD SPAD <1>, BREG ; CLEAR C BIT.
11954		000054	MICPC=MICPC+1
11955	030672	060401	.WORD .S ! DO
11956	030674		MOVE MEM, SPAD <0> MARINC ; GET THE FIRST OPERAND.
11957		000055	MICPC=MICPC+1
11958	030674	057220	.WORD .S
11959	030676		SUBC MEM, SPAD <0>, BR.SP, MARINC ; SPAD <0> := \$FUNC.
11960		000056	MICPC=MICPC+1
11961	030676	057440	.WORD .S
11962	030700		\$IFEQ MEM, SPAD <0> 3S ; BRANCH IF GOOD.
11963			
11964	030700		SUBC SPAD <0>, MEM, NOP
11965		000057	MICPC=MICPC+1
11966	030700	040360	.WORD .S
11967	030702		BZ 3S
11968		000060	MICPC=MICPC+1
11969	030702	131473	.WORD .S
11970	030704		MOVE MEM, OUT1 <4> ; GOOD DATA
11971		000061	MICPC=MICPC+1
11972	030704	041224	.WORD .S
11973	030706		MOVE BREG, OUT1 <5> ; BAD DATA.
11974		000062	MICPC=MICPC+1
11975	030706	061225	.WORD .S
11976	030710		MOVE # 15, BREG ; SET TYPE OF ERROR.
11977		000063	MICPC=MICPC+1
11978	030710	000415	.WORD .S
11979	030712		MOVE BREG, OUT1 <3> ; SET TYPE OF ERROR.
11980		000064	MICPC=MICPC+1
11981	030712	061223	.WORD .S
11982	030714		MOVE # 02, BREG ; LOAD FUNCTION CODE...
11983		000065	MICPC=MICPC+1
11984	030714	000402	.WORD .S
11985	030716		MOVE BREG, OUT1 <CSR7> ; LOAD IT...
11986		000066	MICPC=MICPC+1
11987	030716	061227	.WORD .S
11988	030720		CALL EROR ; ALU SUB W/C ERROR...
11989	030720		MOVE # <MICPC+3>, BREG
11990		000067	MICPC=MICPC+1
11991	030720	000471	.WORD .S
11992	030722		SBR EROR

```

11988      000070      MICPC=MICPC+1
11989      104400      .WORD $
11990      030722      MOVE SPAD (4),MLR ;RESET DATA POINTER...
11991      030724      MICPC=MICPC+1
11992      070204      .WORD $
11993      030726      SBR 25 ;LOOP ON ERROR...
11994      000072      MICPC=MICPC+1
11995      100453      .WORD $
11996      030726      3S: CALL SCP1
11997      030730      MOVE # (MICPC+3),BREG
11998      030730      MICPC=MICPC+1
11999      000475      .WORD $
12000      030732      SBR SCP1
12001      000074      MICPC=MICPC+1
12002      104427      .WORD $
12003      030732      MOVE SPAD (4),MLR ;
12004      030734      MICPC=MICPC+1
12005      070204      .WORD $
12006      030736      SBR 25 ; SCOPE THE DATA....
12007      000076      MICPC=MICPC+1
12008      100453      .WORD $
12009      030740      6S: MOVE SPAD (4),MLR ;RESET DATA POINTER...
12010      000077      MICPC=MICPC+1
12011      070204      .WORD $
12012      030742      MOVE MEM,SPAD (0),MARINC ;GET FIRST OPRAND...
12013      000100      MICPC=MICPC+1
12014      057220      .WORD $
12015      030744      MOVE # 377,BREG ;
12016      000101      MICPC=MICPC+1
12017      030744      .WORD $
12018      030746      SBR SPAD (2),BREG ;SET C BIT...
12019      000102      MICPC=MICPC+1
12020      050402      .WORD $.!.DO
12021      030750      SUBMC MEM,SPAD (0),BR.SP,MARINC ;SP 0 & BR = SUB W/C
12022      000103      MICPC=MICPC+1
12023      057440      .WORD $
12024      030752      MOVE MEM,SPAD (3),MARINC ;DUMMY INSTR, TO MARINC.
12025      000104      MICPC=MICPC+1
12026      057223      .WORD $
12027      030754      SIFEQ MEM,SPAD (0) 9S ;BR IF GOOD...
12028      000105      SUB2C SPAD (0),MEM,NOP
12029      000105      MICPC=MICPC+1
12030      040360      .WORD $
12031      030754      BZ 9S
12032      030756      MICPC=MICPC+1
12033      000106      .WORD $
12034      101521      MOVE MEM,OUT1 (CSR4) ;GOOD DATA.
12035      030760      MICPC=MICPC+1
12036      000107      .WORD $
12037      041224      MOVE BREG,OUT1 (CSRS) ;BAD DATA.
12038      030760      MICPC=MICPC+1
12039      030762      .WORD $
12040      000110      MOVE # 23,BREG ;ERROR TYPE...
12041      061225      MICPC=MICPC+1
12042      030764      .WORD $
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 B 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 B <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 B ;LOOP ON ERROR...
12065 000120 MICPC=MICPC+1
12066 031002 100477 .WORD .S.
12067 031004 9S: CALL SCP1 ;SCOPE THE ERROR...
12068 031004 MOVE B <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 B ;SCOPE THE DATA...
12078 000124 MICPC=MICPC+1
12079 031012 100477 .WORD .S.
12080 031014 MOVE B 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 4S: CALL SCPE ;SCOPE THE TEST...
12095 031026 MOVE B <MICPC+3>,BREG
12096 000131 MICPC=MICPC+1
12097 031026 000533 .WORD .S.
12098 031030 SBR SCPE
12099 000132 MICPC=MICPC+1

```

DZKCA MACY11 27(1006) 13-MAY-77 14:07 PAGE 223
DZKCA.P11 13-MAY-77 13:58 KMC11 ALU TESTS

12100 031030 104454
12101 031032
12102
12103 031032 000133
12104 031034 100400
12105 031034
12106
12107
12108
12109
12110
12111
12112
12113
12114
12115 031034
12116
12117
12118 031034
12119
12120
12121 031034 012737 000055 001202
12122 031042 012737 031362 001442
12123
12124 031050 004737 035536
12125 031054 031070
12126 031056 104022
12127 031060 012706 001200
12128 031064 000177 150352
12129 031070
12130 031070
12131 000000
12132 031070 010000
12133 031072
12134 000001
12135 031072 004000
12136 031074
12137 000002
12138 031074 000400
12139 031076
12140 000003
12141 031076 063236
12142 031100
12143 000004
12144 031100 063220
12145 031102
12146 000005
12147 031102 063221
12148 031104
12149 000006
12150 031104 063222
12151 031106
12152 000007
12153 031106 063162
12154 031110
12155 000010

```
.WORD .S.  
SBR 15 ;DO THE NEXT ITERATION...  
MICPC=MICPC+1  
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  
SXZ  
  
:***** TEST 55 *****  
:ALU TEST  
:TEST OF ALU FUNCTION INC A WITH C BIT CLEARED.  
:TEST OF ALU FUNCTION INC A WITH C BIT SET.  
:ALU FUNCTION (A PLUS 1)  
:LOAD MAIN MEMORY 16 WORDS OF DATA.  
:PERFORM THE FUNCTION, VERIFY THE RESULTS..  
SXZ  
:*****  
STSTN  
: TEST 55  
-----  
TST55: MOV #55,STSTNM ; LOAD THE NO. OF THIS TEST  
MOV #STST55,NEXT ; POINT TO THE START OF NEXT TEST.  
;R1 CONTAINS BASE KMC11 ADDRESS  
;LOAD-VERIFY-WAIT.  
JSR PC,LDRWMT  
MCT55 ERROR 22 ;TIME OUT ERROR...  
MOV #STACK,SP ;RESET STACK.  
JMP @NEXT ;GO TO NEXT TEST...  
MCT55: 15: MOVE #0,MLR ;SET MAR+LO.  
MICPC=MICPC+1  
.WORD .S.  
MOVE #0,MPR ;SET MAR+HI.  
MICPC=MICPC+1  
.WORD .S.  
MOVE #0,BREG ;  
MICPC=MICPC+1  
.WORD .S.  
MOVE BREG,SPAD (16) ;FOR RETURN ADDRESS...  
MICPC=MICPC+1  
.WORD .S.  
MOVE BREG,SPAD (0) ;  
MICPC=MICPC+1  
.WORD .S.  
MOVE BREG,SPAD (1) ;  
MICPC=MICPC+1  
.WORD .S.  
MOVE BREG,SPAD (2) ;  
MICPC=MICPC+1  
.WORD .S.  
SDEC SPAD (2) ;  
MICPC=MICPC+1  
.WORD .S.!.DSP  
MOVE BREG,SPAD (4) ;  
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 ERROR ;ALU INC A ERROR...
12301 031246          MOVE # <MICPC+3>,BREG
12302          000067          MICPC=MICPC+1
12303 031246 000471          .WORD .S.
12304 031250          SBR ERROR
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          NICPC=NICPC+1
12325 031264 100453          .WORD .S.
12326 031266          65: MOVE SPAD <4>,MLR ;RESET DATA POINTER...
12327          000077          NICPC=NICPC+1
12328 031266 070204          .WORD .S.
12329 031270          MOVE MEM,SPAD <0>,MARINC ;GET FIRST OPRAND...
12330          000100          NICPC=NICPC+1
12331 031270 057220          .WORD .S.
12332 031272          MOVE # 377,BREG ;
12333          000101          NICPC=NICPC+1
12334 031272 000777          .WORD .S.
12335 031274          SADD SPAD <2>,BREG ;SET C BIT...
12336          000102          NICPC=NICPC+1
12337 031274 060402          .WORD .S.
12338 031276          SINC SPAD <0>,BR.SP,MARINC ;SP 0 & BR = INC A
12339          000103          NICPC=NICPC+1
12340 031276 077460          .WORD .S.
12341 031300          MOVE MEM,SPAD <3>,MARINC ;DUMMY INSTR, TO MARINC.
12342          000104          NICPC=NICPC+1
12343 031300 057223          .WORD .S.
12344 031302          SIFEQ MEM,SPAD <0> 95 ;BR IF GOOD...
12345
12346
12347 031302          SUB2C SPAD <0>,MEM,NOP
12348          000105          NICPC=NICPC+1
12349 031302 040360          .WORD .S.
12350 031304          BZ 95
12351          000106          NICPC=NICPC+1
12352 031304 101521          .WORD .S.
12353 031306          MOVE MEM,OUT1 <CSR4> ;GOOD DATA.
12354          000107          NICPC=NICPC+1
12355 031306 041224          .WORD .S.
12356 031310          MOVE BREG,OUT1 <CSR5> ;BAD DATA.
12357          000110          NICPC=NICPC+1
12358 031310 061225          .WORD .S.
12359 031312          MOVE # 23,BREG ;ERROR TYPE...
12360          000111          NICPC=NICPC+1
12361 031312 000423          .WORD .S.
12362 031314          MOVE BREG,OUT1 <CSR3> ;
12363          000112          NICPC=NICPC+1
12364 031314 061223          .WORD .S.
12365 031316          MOVE # 03,BREG ;ALU FUNCTION CODE.
12366          000113          NICPC=NICPC+1
12367 031316 000403          .WORD .S.
12368 031320          MOVE BREG,OUT1 <CSR7> ;LOAD IT...
12369          000114          NICPC=NICPC+1
12370 031320 061227          .WORD .S.
12371 031322          CALL EROR ;REPORT ERROR...
12372 031322          MOVE # <NICPC+3>,BREG
12373          000115          NICPC=NICPC+1
12374 031322 000517          .WORD .S.
12375 031324          SBR EROR
12376          000116          NICPC=NICPC+1
12377 031324 104400          .WORD .S.
12378 031326          MOVE SPAD <4>,MLR ;RESTORE DATA POINTER.
12379          000117          NICPC=NICPC+1
    
```

```

12380 031326 070204      .WORD      $
12381 031330                SBR      6$                ;LOOP ON ERROR...
12382                MICPC=MICPC+1
12383 031330 100477      .WORD      $
12384 031330 100477      95:      CALL      SCP1                ;SCOPE THE ERROR...
12385 031332                MOVE      # <MICPC+3>,BREG
12386                MICPC=MICPC+1
12387 031332 000121      .WORD      $
12388 031334 000523      SBR      5$
12389                MICPC=MICPC+1
12390 031334 104427      .WORD      $
12391 031336                MOVE      SPAD <4>,MLR                ;RESTORE DATA POINTER...
12392                MICPC=MICPC+1
12393 031336 000123      .WORD      $
12394 031340                SBR      6$                ;SCOPE THE DATA...
12395                MICPC=MICPC+1
12396 031340 100477      .WORD      $
12397 031342                MOVE      # 4,BREG                ;UPDATE BACKGROUND POINTER.
12398                MICPC=MICPC+1
12399 031342 000125      .WORD      $
12400 031344                SADD     BREG,SPAD <4>,MARINC                ;ALSO DATA POINTER.
12401 031344 077004      .WORD      $,MARINC!.DO
12402 031346                SDEC     SPAD <7>                ;IS IT DONE??
12403                MICPC=MICPC+1
12404 031346 063167      .WORD      $.!.DSP
12405 031350                BZ      4$                ;YES, SCOPE THE TEST.
12406                MICPC=MICPC+1
12407 031350 000127      .WORD      $
12408 031352                SBR      2$                ;DO, THE NEXT.
12409                MICPC=MICPC+1
12410 031352 100453      .WORD      $
12411 031354 45:      CALL      SCOPE                ;SCOPE THE TEST...
12412 031354                MOVE      # <MICPC+3>,BREG
12413                MICPC=MICPC+1
12414 031354 000131      .WORD      $
12415 031356                SBR      SCOPE
12416                MICPC=MICPC+1
12417 031356 000132      .WORD      $
12418 031360                SBR      1$                ;DO THE NEXT ITERATION...
12419                MICPC=MICPC+1
12420 031360 100400      .WORD      $
12421 031362      $ALUT1 0,<2A>,$ASL,0,0,376,376,0,0,376,376,252,252,124,124,252,252,124,124,<A PLUS A>,0
12422 031362      $XZ
12423
12424
12425                ;***** TEST 56 *****
12426                ;*ALU TEST
12427                ;*TEST OF ALU FUNCTION 2A WITH C BIT CLEARED.
12428                ;*TEST OF ALU FUNCTION 2A WITH C BIT SET.
12429                ;*ALU FUNCTION (A PLUS A)
12430                ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
12431                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
12432 031362      $XZ
12433
12434
12435 031362      $STSN

```

```

; TEST 56
12436 ;
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.
;R1 CONTAINS BASE KMC11 ADDRESS
12440 ;
12441 031376 004737 035536 JSR PC,LDRVMT ;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
12447 031416 MCT56:
1S: MOVE #0,MLR ;SET MAR+LO.
MICPC=MICPC+1
.WORD .S.
12448 000000
12449 031416 010000 .WORD .S.
12450 031420 MOVE #0,MPR ;SET MAR+HI.
MICPC=MICPC+1
.WORD .S.
12451 000001
12452 031420 004000 MOVE #0,BREG ;
MICPC=MICPC+1
.WORD .S.
12453 031422 000400 MOVE BREG,SPAD (16) ;FOR RETURN ADDRESS...
MICPC=MICPC+1
.WORD .S.
12454 000002
12455 031422 000400 MOVE BREG,SPAD (0) ;
MICPC=MICPC+1
.WORD .S.
12456 031424 000003 MOVE BREG,SPAD (1) ;
MICPC=MICPC+1
.WORD .S.
12457 000003
12458 031424 063236 MOVE BREG,SPAD (2) ;
MICPC=MICPC+1
.WORD .S.
12459 031426 000004 MOVE BREG,SPAD (2) ;
MICPC=MICPC+1
.WORD .S.
12460 000004
12461 031426 063220 SDEC SPAD (2) ;
MICPC=MICPC+1
.WORD .S.
12462 031430 000005 .S.!DSP
12463 031430 063221 MOVE BREG,SPAD (4) ;
MICPC=MICPC+1
.WORD .S.
12464 031432 000006 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD .S.
12465 031432 063222 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY
MICPC=MICPC+1
.WORD .S.
12466 031432 063222 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY
MICPC=MICPC+1
.WORD .S.
12467 031432 063222 MOVE #0, MEM MARINC ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD .S.
12468 031434 000007 .WORD .S.
12469 000007
12470 031434 063162 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD .S.
12471 031436 000010 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD .S.
12472 000010
12473 031436 063224 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD .S.
12474 031440 000011 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD .S.
12475 000011
12476 031440 016400 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD .S.
12477 031442 000012 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD .S.
12478 000012
12479 031442 016400 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD .S.
12480 031444 000013 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD .S.
12481 000013
12482 031444 016400 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD .S.
12483 031446 000014 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD .S.
12484 000014
12485 031446 016400 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD .S.
12486 031450 000015 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD .S.
12487 000015
12488 031450 016777 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD .S.
12489 031452 000016 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD .S.
12490 000016
12491 031452 016400 MOVE #0, MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD .S.

```

12498	031454	000017	MOVE # 376, MEM MARINC	
12499		016776	MICPC=MICPC+1	
12500	031454		.WORD .S.	
12501	031456		MOVE # 376, MEM MARINC	;RESULT WITH C BIT SET.
12502		000020	MICPC=MICPC+1	
12503	031456	016776	.WORD .S.	
12504	031460		MOVE # 0, MEM MARINC	;LOAD THE DATA IN MEMORY.
12505		000021	MICPC=MICPC+1	
12506	031460	016400	.WORD .S.	
12507	031462		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
12508		000022	MICPC=MICPC+1	
12509	031462	016777	.WORD .S.	
12510	031464		MOVE # 0, MEM MARINC	
12511		000023	MICPC=MICPC+1	
12512	031464	016400	.WORD .S.	
12513	031466		MOVE # 0, MEM MARINC	;RESULT WITH C BIT SET.
12514		000024	MICPC=MICPC+1	
12515	031466	016400	.WORD .S.	
12516	031470		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY
12517		000025	MICPC=MICPC+1	
12518	031470	016777	.WORD .S.	
12519	031472		MOVE # -1, MEM MARINC	;LOAD THE DATA IN MEMORY.
12520		000026	MICPC=MICPC+1	
12521	031472	016777	.WORD .S.	
12522	031474		MOVE # 376, MEM MARINC	
12523		000027	MICPC=MICPC+1	
12524	031474	016776	.WORD .S.	
12525	031476		MOVE # 376, MEM MARINC	;RESULT WITH C BIT SET.
12526		000030	MICPC=MICPC+1	
12527	031476	016776	.WORD .S.	
12528	031500		MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
12529		000031	MICPC=MICPC+1	
12530	031500	016525	.WORD .S.	
12531	031502		MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
12532		000032	MICPC=MICPC+1	
12533	031502	016525	.WORD .S.	
12534	031504		MOVE # 252, MEM MARINC	
12535		000033	MICPC=MICPC+1	
12536	031504	016652	.WORD .S.	
12537	031506		MOVE # 252, MEM MARINC	;RESULT WITH C BIT SET.
12538		000034	MICPC=MICPC+1	
12539	031506	016652	.WORD .S.	
12540	031510		MOVE # 252, MEM MARINC	;LOAD THE DATA IN MEMORY.
12541		000035	MICPC=MICPC+1	
12542	031510	016652	.WORD .S.	
12543	031512		MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
12544		000036	MICPC=MICPC+1	
12545	031512	016525	.WORD .S.	
12546	031514		MOVE # 124, MEM MARINC	
12547		000037	MICPC=MICPC+1	
12548	031514	016524	.WORD .S.	
12549	031516		MOVE # 124, MEM MARINC	;RESULT WITH C BIT SET.
12550		000040	MICPC=MICPC+1	
12551	031516	016524	.WORD .S.	
12552	031520		MOVE # 125, MEM MARINC	;LOAD THE DATA IN MEMORY.
12553		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		SACD 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> 3\$;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 3\$	
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 ERROR	;ALU 2A ERROR...
12618	031574		MOVE # <MICPC+3>,BREG	
12619		000067	MICPC=MICPC+1	
12620	031574	000471	.WORD .S.	
12621	031576		SBR ERROR	
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.:DL	
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	

35:

65:

```

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 000105 MICPC=MICPC+1
12666 031630 040360 .WORD .S.
12667 031632 BZ 9$
12668 000106 MICPC=MICPC+1
12669 031632 101521 .WORD .S.
12670 031634 MOVE MEM,OUT1 <CSR4> ;GOOD DATA.
12671 000107 MICPC=MICPC+1
12672 031634 041224 .WORD .S.
12673 031636 MOVE BREG,OUT1 <CSR5> ;BAD DATA.
12674 000110 MICPC=MICPC+1
12675 031636 061225 .WORD .S.
12676 031640 MOVE # 23,BREG ;ERROR TYPE...
12677 000111 MICPC=MICPC+1
12678 031640 000423 .WORD .S.
12679 031642 MOVE BREG,OUT1 <CSR3> ;
12680 000112 MICPC=MICPC+1
12681 031642 061223 .WORD .S.
12682 031644 MOVE # 05,BREG ;ALU FUNCTION CODE.
12683 000113 MICPC=MICPC+1
12684 031644 000405 .WORD .S.
12685 031646 MOVE BREG,OUT1 <CSR7> ;LOAD IT...
12686 000114 MICPC=MICPC+1
12687 031646 061227 .WORD .S.
12688 031650 CALL EROR ;REPORT ERROR...
12689 031650 MOVE # <MICPC+3>,BREG
12690 000115 MICPC=MICPC+1
12691 031650 000517 .WORD .S.
12692 031652 SBR EROR
12693 000116 MICPC=MICPC+1
12694 031652 104400 .WORD .S.
12695 031654 MOVE SPAD <4>,MLR ;RESTORE DATA POINTER.
12696 000117 MICPC=MICPC+1
12697 031654 070204 .WORD .S.
12698 031656 SBR 6$ ;LOOP ON ERROR...
12699 000120 MICPC=MICPC+1
12700 031656 100477 .WORD .S.
12701 031660 9$: CALL SCP1 ;SCOPE THE ERROR...
12702 031660 MOVE # <MICPC+3>,BREG
12703 000121 MICPC=MICPC+1
12704 031660 000523 .WORD .S.
12705 031662 SBR SCP1
12706 000122 MICPC=MICPC+1
12707 031662 104427 .WORD .S.
12708 031664 MOVE SPAD <4>,MLR ;RESTORE DATA POINTER...
12709 000123 MICPC=MICPC+1
12710 031664 070204 .WORD .S.
12711 031666 SBR 6$ ;SCOPE THE DATA...
12712 000124 MICPC=MICPC+1
12713 031666 100477 .WORD .S.
12714 031670 MOVE # 4,BREG ;UPDATE BACKGROUND POINTER.
12715 000125 MICPC=MICPC+1
    
```

```

12716 031670 000404      WORD      $      BREG SPAD (4), MARINC      ;ALSO DATA POINTER.
12717 031672      $A00      $      !MARINC!.DO
12718 031672 077004      .WORD     $      SPAD (7)      ;IS IT DONE??
12719 031674      SDEC      $      SPAD (7)
12720      000126      MICPC=MICPC+1
12721 031674 063167      .WORD     $      !.DSP
12722 031676      BZ      4$      ;YES, SCOPE THE TEST.
12723      000127      MICPC=MICPC+1
12724 031676 101532      .WORD     $
12725 031700      SBR      2$      ;DO, THE NEXT.
12726      000130      MICPC=MICPC+1
12727 031700 100453      .WORD     $
12728 031702 4$:      CALL     SCPE      ;SCOPE THE TEST...
12729 031702      MOVE     # (MICPC+3),BREG
12730      000131      MICPC=MICPC+1
12731 031702 000533      .WORD     $
12732 031704      SBR      SCPE
12733      000132      MICPC=MICPC+1
12734 031704 104454      .WORD     $
12735 031706      SBR      1$      ;DO THE NEXT ITERATION...
12736      000133      MICPC=MICPC+1
12737 031706 100403      .WORD     $
12738 031710 SALUT1 0, (A PLUS C), $ADC, 0, 1, -1, 0, 0, 1, -1, 0, 125, 126, 252, 253, 125, 126, 252, 253, (A PLUS C), 0
12739 031710 $XZ
12740
12741
12742      ;***** TEST 57 *****
12743      ;#ALU TEST
12744      ;#TEST OF ALU FUNCTION A PLUS C WITH C BIT CLEARED.
12745      ;#TEST OF ALU FUNCTION A PLUS C WITH C BIT SET.
12746      ;#ALU FUNCTION (A PLUS C)
12747      ;#LOAD MAIN MEMORY 16 WORDS OF DATA.
12748      ;#PERFORM THE FUNCTION, VERIFY THE RESULTS..
12749 031710 $XZ
12750
12751      ;*****
12752 031710 $TSTN
12753      ; TEST 57
12754
12755 031710 012737 000057 001202 TST57:  MOV     #57,$TSTN      ; LOAD THE N THIS TEST
12756 031716 012737 032236 001442  MOV     #ST160,NEXT   ; POINT TO 1 RT OF NEXT TEST.
12757      JSR     PC,LDRMNT   ;R1 CONTAINS BASE K... ADDRESS
12758 031724 004737 035536      MCT57  ERROR      22      ;LOAD-VERIFY-WAIT.
12759 031730 031744      ERROR      22      ;TIME OUT ERROR...
12760 031732 104022      MOV     #STACK,SP    ;RESET STACK...
12761 031734 012706 001200      JMP     @NEXT        ;GO TO NEXT TEST...
12762 031740 000177 147476      MCT57:  IS:      MOVE     # 0,MLR      ;SET MAR+LO.
12763 031744      MICPC=MICPC+1
12764      .WORD     $
12765      000000      MOVE     # 0,MPR      ;SET MAR+HI.
12766 031744 010000      MICPC=MICPC+1
12767 031746      .WORD     $
12768      000001      MOVE     # 0,BREG
12769 031746 004000      MICPC=MICPC+1
12770 031750      .WORD     $
12771      000002

```

12772	031750	000400	.WORD	.S.		
12773	031752		MOVE	BREG SPAD (16)		;FOR RETURN ADDRESS...
12774		000003	NICPC=NICPC+1			
12775	031752	063236	.WORD	.S.		
12776	031754		MOVE	BREG SPAD (0)		;
12777		000004	NICPC=NICPC+1			
12778	031754	063220	.WORD	.S.		
12779	031756		MOVE	BREG SPAD (1)		;
12780		000005	NICPC=NICPC+1			
12781	031756	063221	.WORD	.S.		
12782	031760		MOVE	BREG SPAD (2)		;
12783		000006	NICPC=NICPC+1			
12784	031760	063222	.WORD	.S.		
12785	031762		SDC	SPAD (2)		;
12786		000007	NICPC=NICPC+1			
12787	031762	063162	.WORD	.S.!.DSP		
12788	031764		MOVE	BREG SPAD (4)		;
12789		000010	NICPC=NICPC+1			
12790	031764	063224	.WORD	.S.		
12791	031766		MOVE	# 0 MEM MARINC		;LOAD THE DATA IN MEMORY.
12792		000011	NICPC=NICPC+1			
12793	031766	016400	.WORD	.S.		
12794	031770		MOVE	# 0 MEM MARINC		;LOAD THE DATA IN MEMORY
12795		000012	NICPC=NICPC+1			
12796	031770	016400	.WORD	.S.		
12797	031772		MOVE	# 0 MEM MARINC		
12798		000013	NICPC=NICPC+1			
12799	031772	016400	.WORD	.S.		
12800	031774		MOVE	# 1 MEM MARINC		;RESULT WITH C BIT SET.
12801		000014	NICPC=NICPC+1			
12802	031774	016401	.WORD	.S.		
12803	031776		MOVE	# -1 MEM MARINC		;LOAD THE DATA IN MEMORY.
12804		000015	NICPC=NICPC+1			
12805	031776	016777	.WORD	.S.		
12806	032000		MOVE	# 0 MEM MARINC		;LOAD THE DATA IN MEMORY.
12807		000016	NICPC=NICPC+1			
12808	032000	016400	.WORD	.S.		
12809	032002		MOVE	# -1 MEM MARINC		
12810		000017	NICPC=NICPC+1			
12811	032002	016777	.WORD	.S.		
12812	032004		MOVE	# 0 MEM MARINC		;RESULT WITH C BIT SET.
12813		000020	NICPC=NICPC+1			
12814	032004	016400	.WORD	.S.		
12815	032006		MOVE	# 0 MEM MARINC		;LOAD THE DATA IN MEMORY.
12816		000021	NICPC=NICPC+1			
12817	032006	016400	.WORD	.S.		
12818	032010		MOVE	# -1 MEM MARINC		;LOAD THE DATA IN MEMORY.
12819		000022	NICPC=NICPC+1			
12820	032010	016777	.WORD	.S.		
12821	032012		MOVE	# 0 MEM MARINC		
12822		000023	NICPC=NICPC+1			
12823	032012	016400	.WORD	.S.		
12824	032014		MOVE	# 1 MEM MARINC		;RESULT WITH C BIT SET.
12825		000024	NICPC=NICPC+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		2S: 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	.WORD .S.	
12943	032126	000071	MOVE SPAD <4>,MLR	;RESET DATA POINTER...
12944	032126	070204	MICPC=MICPC+1	
12945	032130	000072	.WORD .S.	
12946	032130	100453	SBR	;LOOP ON ERROR...
12947	032132		MICPC=MICPC+1	
12948	032132		.WORD .S.	
12949	032132	000073	CALL SCP1	
12950	032132	000475	MOVE # <MICPC+3>,BREG	
12951	032134		MICPC=MICPC+1	
12952	032134	000074	.WORD .S.	
12953	032134	104427	SBR SCP1	
12954	032136		MICPC=MICPC+1	
12955	032136	000075	.WORD .S.	
12956	032136	070204	MOVE SPAD <4>,MLR	;
12957	032140		MICPC=MICPC+1	
12958	032140	000076	.WORD .S.	
12959	032140	100453	SBR	;SCOPE THE DATA....
12960	032142		MICPC=MICPC+1	
12961	032142	000077	.WORD .S.	
12962	032142	070204	MOVE SPAD <4>,MLR	;RESET DATA POINTER...
12963	032144		MICPC=MICPC+1	
12964	032144	000100	MOVE MEM,SPAD <0>,MARINC	;GET FIRST OPRAND...
12965	032144	057220	MICPC=MICPC+1	
12966	032146		.WORD .S.	
12967	032146	000101	MOVE # 377,BREG	;
12968	032146	000777	MICPC=MICPC+1	
12969	032150		.WORD .S.	
12970	032150	000102	SADD SPAD <2>,BREG	;SET C BIT...
12971	032150	060402	MICPC=MICPC+1	
12972	032152		.WORD .S.!.00	
12973	032152	000103	SADC SPAD <0>,BR.SP,MARINC	;SP 0 & BR = A PLUS C
12974	032152	077500	MICPC=MICPC+1	
12975	032154		.WORD .S.	
12976	032154	000104	MOVE MEM,SPAD <3>,MARINC	;DUMMY INSTR, TO MARINC.
12977	032154	057223	MICPC=MICPC+1	
12978	032156		.WORD .S.	
12979	032156		SIFE0	95 ;BR IF GOOD...
12980	032156			
12981	032156	000105	SUB2C	
12982	032156	040360	MICPC=MICPC+1	
12983	032160		.WORD .S.	
12984	032160		BZ 95	
12985	032160	000106	MICPC=MICPC+1	
12986	032160	101521	.WORD .S.	
12987	032162		MOVE MEM,OUT1 <CSR4>	;GOOD DATA.
12988	032162	000107	MICPC=MICPC+1	
12989	032162	041224	.WORD .S.	
12990	032164		MOVE BREG,OUT1 <CSR5>	;BAD DATA.
12991	032164	000110	MICPC=MICPC+1	
12992	032164	061225	.WORD .S.	
12993	032166		MOVE # 23,BREG	;ERROR TYPE...
12994	032166	000111	MICPC=MICPC+1	
12995	032166	000423	.WORD .S.	

```

12996 032170      MOVE      BREG,OUT1 <CSR3>          ;
12997          000112      NICPC=NICPC+1
12998 032170      .WORD      .S.
12999 032172      MOVE      # 04,BREG          ;ALU FUNCTION CODE.
13000          000113      NICPC=NICPC+1
13001 032172      .WORD      .S.
13002 032174      MOVE      BREG,OUT1 <CSR7>      ;LOAD IT...
13003          000114      NICPC=NICPC+1
13004 032174      .WORD      .S.
13005 032176      CALL      EROR,REPORT ERROR...
13006 032176      MOVE      # <NICPC+3>,BREG
13007          000115      NICPC=NICPC+1
13008 032176      .WORD      .S.
13009 032200      SBR      EROR
13010          000116      NICPC=NICPC+1
13011 032200      .WORD      .S.
13012 032202      MOVE      SPAD <4>,MLR          ;RESTORE DATA POINTER.
13013          000117      NICPC=NICPC+1
13014 032202      .WORD      .S.
13015 032204      SBR      #S          ;LOOP ON ERROR...
13016          000120      NICPC=NICPC+1
13017 032204      .WORD      .S.
13018 032206      95:      CALL      SCPE          ;SCOPE THE ERROR...
13019 032206      MOVE      # <NICPC+3>,BREG
13020          000121      NICPC=NICPC+1
13021 032206      .WORD      .S.
13022 032210      SBR      SCPE
13023          000122      NICPC=NICPC+1
13024 032210      .WORD      .S.
13025 032212      MOVE      SPAD <4>,MLR          ;RESTORE DATA POINTER...
13026          000123      NICPC=NICPC+1
13027 032212      .WORD      .S.
13028 032214      SBR      #S          ;SCOPE THE DATA...
13029          000124      NICPC=NICPC+1
13030 032214      .WORD      .S.
13031 032216      MOVE      # 4,BREG          ;UPDATE BACKGROUND POINTER.
13032          000125      NICPC=NICPC+1
13033 032216      .WORD      .S.
13034 032220      SADD      BREG,SPAD <4>,MARINC      ;ALSO DATA POINTER.
13035 032220      .WORD      .S,!MARINC!.00
13036 032222      SDEC      SPAD <7>          ;IS IT DONE??
13037          000126      NICPC=NICPC+1
13038 032222      .WORD      .S,!DSP
13039 032224      BZ      #4S          ;YES, SCOPE THE TEST.
13040          000127      NICPC=NICPC+1
13041 032224      .WORD      .S.
13042 032226      SBR      #S          ;DO, THE NEXT.
13043          000130      NICPC=NICPC+1
13044 032226      .WORD      .S.
13045 032230      45:      CALL      SCPE          ;SCOPE THE TEST...
13046 032230      MOVE      # <NICPC+3>,BREG
13047          000131      NICPC=NICPC+1
13048 032230      .WORD      .S.
13049 032232      SBR      SCPE
13050          000132      NICPC=NICPC+1
13051 032232      .WORD      .S.
    
```

```

13052 032234          SBR      15          ;DO THE NEXT ITERATION...
13053          000133  MICPC=MICPC+1
13054 032234          WORD      .S.
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,LDRANT       ;LOAD-VERIFY-WAIT.
13076 032256 032272 MCT60
13077 032260 104022 ERROR
13078 032262 012706 001200 MOV      #STACK,SP       ;TIME OUT ERROR...
13079 032266 000177 147150 JMP      @NEXT           ;RESET STACK...
13080 032272          MCT60:          ;GO TO NEXT TEST...
13081 032272          15:          MOVE      #0,MLR          ;SET MAR+LO.
13082          MICPC=MICPC+1
13083          .WORD      .S.
13084 032274          MOVE      #0,MPR          ;SET MAR+HI.
13085          MICPC=MICPC+1
13086 032274 004000          .WORD      .S.
13087 032276          MOVE      #0,BREG
13088          MICPC=MICPC+1          ;
13089 032276 000400          .WORD      .S.
13090 032300          MOVE      BREG,SPAD (16)      ;FOR RETURN ADDRESS...
13091          MICPC=MICPC+1
13092 032300 063236          .WORD      .S.
13093 032302          MOVE      BREG,SPAD (0)          ;
13094          MICPC=MICPC+1
13095 032302 063220          .WORD      .S.
13096 032304          MOVE      BREG,SPAD (1)          ;
13097          MICPC=MICPC+1
13098 032304 063221          .WORD      .S.
13099 032306          MOVE      BREG,SPAD (2)          ;
13100          MICPC=MICPC+1
13101 032306 063222          .WORD      .S.
13102 032310          SDEC      SPAD (2)          ;
13103          MICPC=MICPC+1
13104 032310 063162          .WORD      .S.! .DSP
13105 032312          MOVE      BREG,SPAD (4)          ;
13106          MICPC=MICPC+1
13107 032312 063224          .WORD      .S.

```

JZKCA MACY11 27(1006) 13-MAY-77 14:07 PAGE 241
 JZKCA.P11 13-MAY-77 13:58 KMC11 ALU TESTS

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

MACY11 27(1006) 13-MAY-77 14:07 PAGE 242
 4 P11 13-MAY-77 13:58 KMC11 ALU TESTS

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	032354		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		25: 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		\$IFEQ	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	

35:

13276	032466	100453	
13277	032470		65: .WORD \$
13278		000077	MOVE SPAD <4>,MLR ;RESET DATA POINTER...
13279	032470	070204	MICPC=MICPC+1
13280	032472		.WORD \$
13281		000100	MOVE MEM,SPAD <0>,MARINC ;GET FIRST OPRAND...
13282	032472	057220	MICPC=MICPC+1
13283	032474		.WORD \$
13284		000101	MOVE # 377,BREG ;
13285	032474	000777	MICPC=MICPC+1
13286	032476		.WORD \$
13287		000102	SADD SPAD <2>,BREG ;SET C BIT...
13288	032476	060402	MICPC=MICPC+1
13289	032500		.WORD \$
13290		000103	SUBC MEM,SPAD <0>,BR.SP,MARINC ;SP 0 & BR = 2'S COMP SUB
13291	032500	057760	MICPC=MICPC+1
13292	032502		.WORD \$
13293		000104	MOVE MEM,SPAD <3>,MARINC ;DUMMY INSTR, TO MARINC.
13294	032502	057223	MICPC=MICPC+1
13295	032504		.WORD \$
13296			SIFEQ MEM,SPAD <C> 95 ;BR IF GOOD...
13297			
13298	032504		SUBC SPAD <0>,MEM,NOP
13299		000105	MICPC=MICPC+1
13300	032504	040360	.WORD \$
13301	032506		BZ \$
13302		000106	MICPC=MICPC+1
13303	032506	101521	.WORD \$
13304	032510		MOVE MEM,OUT1 <CSR4> ;GOOD DATA.
13305		000107	MICPC=MICPC+1
13306	032510	041224	.WORD \$
13307	032512		MOVE BREG,OUT1 <CSR5> ;BAD DATA.
13308		000110	MICPC=MICPC+1
13309	032512	061225	.WORD \$
13310	032514		MOVE # 23,BREG ;ERROR TYPE...
13311		000111	MICPC=MICPC+1
13312	032514	000423	.WORD \$
13313	032516		MOVE BREG,OUT1 <CSR3> ;
13314		000112	MICPC=MICPC+1
13315	032516	061223	.WORD \$
13316	032520		MOVE # 17,BREG ;ALU FUNCTION CODE.
13317		000113	MICPC=MICPC+1
13318	032520	000417	.WORD \$
13319	032522		MOVE BREG,OUT1 <CSR7> ;LOAD IT...
13320		000114	MICPC=MICPC+1
13321	032522	061227	.WORD \$
13322	032524		CALL EROR ;REPORT ERROR...
13323	032524		MOVE # <MICPC+3>,BREG
13324		000115	MICPC=MICPC+1
13325	032524	000517	.WORD \$
13326	032526		SBR EROR
13327		000116	MICPC=MICPC+1
13328	032526	104400	.WORD \$
13329	032530		MOVE SPAD <4>,MLR ;RESTORE DATA POINTER.
13330		000117	MICPC=MICPC+1
13331	032530	070204	.WORD \$

```

13332 032532          SBR          65          ;LOOP ON ERROR...
13333          MICPC=MICPC+1
13334          .WORD          .S.
13335 032532 000120 95: CALL          SCP1          ;SCOPE THE ERROR...
13336 032534          MOVE          8 <MICPC+3>,BREG
13337 032534          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 070204          .WORD          .S.
13345 032542          SBR          65          ;SCOPE THE DATA...
13346          MICPC=MICPC+1
13347 032542 000124          .WORD          .S.
13348 032544 100477          MOVE          8 4,BREG          ;UPDATE BACKGROUND POINTER.
13349          MICPC=MICPC+1
13350 032544 000125          .WORD          .S.
13351 032546 000404          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          45: CALL          SCOPE          ;SCOPE THE TEST...
13363 032556          MOVE          8 <MICPC+3>,BREG
13364          MICPC=MICPC+1
13365 032556 000131          .WORD          .S.
13366 032560 000533          SBR          SCOPE
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          ;***** TEST 61 *****
13376          ;#ALU TEST
13377          ;#TEST OF ALU FUNCTION DEC A WITH C BIT CLEARED.
13378          ;#TEST OF ALU FUNCTION DEC A WITH C BIT SET.
13379          ;#ALU FUNCTION (A-1)
13380          ;#LOAD MAIN MEMORY 16 WORDS OF DATA.
13381          ;#PERFORM THE FUNCTION, VERIFY THE RESULTS..
13382          SXZ
13383 032564          ;*****
13384
13385          STSTN
13386 032564          ; TEST 61
13387

```

```

13388 032514 012737 000061 001202 TST61: MOV #61,STSTNM ; LOAD THE NO. OF THIS TEST
13389 032572 012737 033112 001442 MOV #TST62,NEXT ; POINT TO THE START OF NEXT TEST.
13391 032600 004737 035536 JSR PC,LVTRWT ;R1 CONTAINS BASE KMC11 ADDRESS
13392 032604 032620 MCT61 ;LOAD-VERIFY-WAIT.
13393 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...
13398 032620 NCT61: 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 000042 MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY
13501 032724 016652 MICPC=MICPC+1
13502 032724 016652 .WORD .S.
13503 032726 MOVE # 124, MEM MARINC
13504 032726 000043 MICPC=MICPC+1
13505 032726 016652 .WORD .S.
13506 032730 MOVE # 124, MEM MARINC ;RESULT WITH C BIT SET.
13507 032730 000044 MICPC=MICPC+1
13508 032730 016652 .WORD .S.
13509 032732 MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
13510 032732 000045 MICPC=MICPC+1
13511 032732 016652 .WORD .S.
13512 032734 MOVE # 252, MEM MARINC ;LOAD THE DATA IN MEMORY.
13513 032734 000046 MICPC=MICPC+1
13514 032734 016652 .WORD .S.
13515 032736 MOVE # 251, MEM MARINC ;LOAD THE DATA IN MEMORY.
13516 032736 000047 MICPC=MICPC+1
13517 032736 016651 .WORD .S.
13518 032740 MOVE # 251, MEM MARINC ;RESULT WITH C BIT SET.
13519 032740 000050 MICPC=MICPC+1
13520 032740 016651 .WORD .S.
13521 032742 MOVE # 7, SPAD <7> ;SET THE COUNT.
13522 032742 000051 MICPC=MICPC+1
13523 032742 003007 .WORD .S.
13524 032744 MOVE # 0, MLR ;MAR+0.
13525 032744 000052 MICPC=MICPC+1
13526 032744 010000 .WORD .S.
13527 032746 000053 25: MOVE # 0, BREG ;
13528 032746 000400 MICPC=MICPC+1
13529 032746 000400 .WORD .S.
13530 032750 $A00 SPAD <1>, BREG ;CLEAR C BIT.
13531 032750 000054 MICPC=MICPC+1
13532 032750 060401 .WORD .S.!.DO
13533 032752 MOVE MEM, SPAD <0> MARINC ;GET THE FIRST OPERAND.
13534 032752 000055 MICPC=MICPC+1
13535 032752 057220 .WORD .S.
13536 032754 $0EC SPAD <0>, BR. SP, MARINC ;
13537 032754 000056 MICPC=MICPC+1
13538 032754 077560 .WORD .S.
13539 032756 $IFEQ MEM, SPAD <0> 3$ ;BRANCH IF GOOD.
13540 032756 000057 SUB2C SPAD <0>, MEM, NOP
13541 032756 0403E0 MICPC=MICPC+1
13542 032760 .WORD .S.
13543 032760 BZ 3$
13544 032760 000060 MICPC=MICPC+1
13545 032760 101473 .WORD .S.
13546 032762 MOVE MEM, OUT1 <4> ;GOOD DATA
13547 032762 000061 MICPC=MICPC+1
13548 032762 041224 .WORD .S.
13549 032764 MOVE BREG, OUT1 <5> ;BAD DATA.
13550 032764 000062 MICPC=MICPC+1
13551 032764 061225 .WORD .S.

```

1355	032766	000063	MOVE # 15, BREG ; SET TYPE OF ERROR.
1356			MICPC=MICPC+1
1357	032766	000415	.WORD .S.
1358	032770		MOVE BREG, OUT1 <3> ; SET TYPE OF ERROR.
1359			MICPC=MICPC+1
1360	032770	000064	.WORD .S.
1361	032772	061223	MOVE # 07, BREG ; LOAD FUNCTION CODE...
1362	032772		MICPC=MICPC+1
1363		000065	.WORD .S.
1364	032772	000407	MOVE BREG, OUT1 <CSR7> ; LOAD IT...
1365	032774		MICPC=MICPC+1
1366		000066	.WORD .S.
1367	032774	061227	CALL ERROR ; ALU DEC A ERROR...
1368	032776		MOVE # <MICPC+3>, BREG
1369	032776		MICPC=MICPC+1
1370		000067	.WORD .S.
1371	032776	000471	SBR ERROR
1372	033000		MICPC=MICPC+1
1373		000070	.WORD .S.
1374	033000	104400	MOVE SPAD <4>, MLR ; RESET DATA POINTER...
1375	033002		MICPC=MICPC+1
1376		000071	.WORD .S.
1377	033002	070204	SBR # 25 ; LOOP ON ERROR...
1378	033004		MICPC=MICPC+1
1379		000072	.WORD .S.
1380	033004	100453	CALL SCP1
1381	033006		MOVE # <MICPC+3>, BREG
1382	033006		MICPC=MICPC+1
1383		000073	.WORD .S.
1384	033006	000475	SBR SCP1
1385	033010		MICPC=MICPC+1
1386		000074	.WORD .S.
1387	033010	104427	MOVE SPAD <4>, MLR ;
1388	033012		MICPC=MICPC+1
1389		000075	.WORD .S.
1390	033012	070204	SBR # 25 ; SCOPE THE DATA....
1391	033014		MICPC=MICPC+1
1392		000076	.WORD .S.
1393	033014	100453	MOVE SPAD <4>, MLR ; RESET DATA POINTER...
1394	033016		MICPC=MICPC+1
1395		000077	.WORD .S.
1396	033016	070204	MOVE # 0, MARINC ; GET FIRST OPRAND...
1397	033020		MICPC=MICPC+1
1398		000100	.WORD .S.
1399	033020	057220	MOVE # 377, BREG ;
1400	033022		MICPC=MICPC+1
1401		000101	.WORD .S.
1402	033022	000777	SADD SPAD <2>, BREG ; SET C BIT...
1403	033024		MICPC=MICPC+1
1404		000102	.WORD .S.
1405	033024	060402	SDEC SPAD <0>, BR.SP, MARINC ; SP 0 & BR = DEC A
1406	033026		MICPC=MICPC+1
1407		000103	.WORD .S.
1408	033026	077560	MOVE # 0, MARINC ; DUMMY INSTR, TO MARINC.
1409	033030		MICPC=MICPC+1
1410		000104	.WORD .S.
1411	033030	057223	

35:

65:

```

13612 033032          SIFE0  MEM,SPAD <0>  95      ;BR IF GOOD...
13613
13614
13615 033032          SUBRC  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   65      ;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   65      ;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          SROO  BREG,SPAD <4>,MARINC ;ALSO DATA POINTER.
13669 033074 077004  .WORD  .S. !MARINC!.00
13670 033076          SDEC  SPAD <7>          ;IS IT DONE??
13671          000126  MICPC=MICPC+1
13672 033076 063167  .WORD  .S.!.DSP
13673 033100          BZ      45          ;YES, SCOPE THE TEST.
13674          000127  MICPC=MICPC+1
13675 033100 101532  .WORD  .S.
13676 033102          SBR      25          ;DO, THE NEXT.
13677          000130  MICPC=MICPC+1
13678 033102 100453  .WORD  .S.
13679 033104 45:    CALL  SCPE          ;SCOPE THE TEST...
13680 033104          MOVE  # <MICPC+3>,BREG
13681          000131  MICPC=MICPC+1
13682 033104 000533  .WORD  .S.
13683 033106          SBR      SCPE
13684          000132  MICPC=MICPC+1
13685 033106 104454  .WORD  .S.
13686 033110          SBR      15          ;DO THE NEXT ITERATION...
13687          000133  MICPC=MICPC+1
13688 033110 100400  .WORD  .S.
13689 033112          SPCHK
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          :
13703          :-----
13704          : TEST 62
13705 033112 012737 000062 001202 TST62:  MOV  #62,STSTN          ; LOAD THE NO OF THIS TEST
13706 033120 012737 033260 001442  MOV  #TST63,NEXT      ; POINT TO THE START OF NEXT TEST.
13707 033126 004737 035536          JSR  PC,LDRWT          ;R1 CONTAINS BASE KMC11 ADDRESS
13708 033132 033146          MCT62          ;LOAD-VERIFY-WAIT.
13709 033134 104022          TROR  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          IS:  MOVE  # 0,BREG          ;PREPARE
13714          000000  MICPC=MICPC+1
13715 033146 000400  .WORD  .S.
13716 033150          MOVE  BREG,SPAD <2> ; FOR
13717          000001  MICPC=MICPC+1
13718 033150 063222  .WORD  .S.
13719 033152          MOVE  BREG,SPAD <3> ; DELAY.
13720          000002  MICPC=MICPC+1
13721 033152 063223  .WORD  .S.
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		BPH 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		BPH 35	;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			

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 5S ;YES, IT'S OK.
MICPC=MICPC+1
.WORD .S.
$INC SPAD <2> ;IN DELAY!
MICPC=MICPC+1
.WORD .S.!.DSP
$ADC SPAD <3> ;DELAY!
MICPC=MICPC+1
.WORD .S.!.DSP
MOVE SPAD <3>,BREG ;IS DELAYED ENOUGH?
MICPC=MICPC+1
.WORD .S.
BB7 6S ;YES, REPORT ERROR.
MICPC=MICPC+1
.WORD .S.
SBR 4S ;CONTINUE DELAY LOOP.
MICPC=MICPC+1
.WORD .S.
6S: MOVE # 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 # <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 SCOPE ;SCOPE THE TEST.
MOVE # <MICPC+3>,BREG
MICPC=MICPC+1
.WORD .S.
SBR SCOPE
MICPC=MICPC+1
.WORD .S.
SBR 1S ;DO NEXT PASS.
MICPC=MICPC+1
.WORD .S.
SNOISE
SXZ
```

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

13836
13837
13838
13839
13840
13841
13842
13843
13844
13845
13846
13847
13848
13849
13850
13851
13852
13853
13854
13855
13856
13857
13858
13859
13860
13861
13862
13863
13864
13865
13866
13867
13868
13869
13870
13871
13872
13873
13874
13875
13876
13877
13878
13879
13880
13881
13882
13883
13884
13885
13886
13887
13888
13889
13890
13891

033260

033260
033266
033274
033300
033302
033304
033310
033314
033314
000000
033314
033316
000001
033316
033320
000002
033320
033322
000003
033322
033324
000004
033324
033326
000005
033326
033330
000006
033330
033332
000007
033332
061222
033334
000010
033334
061223
033336
000011
033336
061224
033340
000012
033340
061225
033342

```

;WRITE ALL ZERO'S THEN ALL ONE'S THEN A DATA PATTERN TO
;THE IBUS, IBUS, SP, & MAIN MEMORY
;THEN GO AND READ BACK THE DATA PATTERN
;TO VERIFY THAT READING AND WRITING OF OTHER
;LOCATIONS DID NOT CHANGE DATA.
SXZ
;*****
STSTN
; TEST 63
TST63: MOV #63,STSTNM ; LOAD THE NO. OF THIS TEST
MOV #TST64,NEXT ; POINT TO THE START OF NEXT TEST.
;R1 CONTAINS BASE KMC11 ADDRESS
;LOAD-VERIFY-WAIT.
.TSR PC,LDVMT
MCT63
ERROR 22 ;TIME OUT ERROR...
MOV #STACK,SP ;RESET STACK...
JMP @NEXT ;GO TO NEXT TEST...
MCT63:
IBS: MOVE #0,BREG ;START WITH ZERO.
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 <16> ;FOR RETURN ADDRESS PURPOSES...
NICPC=NICPC+1
.WORD .S.
IS: MOVE #0,MLR ;MAR+0.
NICPC=NICPC+1
.WORD .S.
MOVE #0,MPR ;
NICPC=NICPC+1
.WORD .S.
MOVE BREG,OUT1 <CSR0> ;WRITE IBUS* REGISTERS.
NICPC=NICPC+1
.WORD .S.
MOVE BREG,OUT1 <CSR2> ;WRITE IBUS* REGISTERS.
NICPC=NICPC+1
.WORD .S.
MOVE BREG,OUT1 <CSR3> ;WRITE IBUS* REGISTERS.
NICPC=NICPC+1
.WORD .S.
MOVE BREG,OUT1 <CSR4> ;WRITE IBUS* REGISTERS.
NICPC=NICPC+1
.WORD .S.
MOVE BREG,OUT1 <CSR5> ;WRITE IBUS* REGISTERS.
NICPC=NICPC+1
.WORD .S.
MOVE BREG,OUT1 <CSR6> ;WRITE IBUS* REGISTERS.

```

```

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

```

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

MACY11 27(1006) 13-MAY-77 14:07 PAGE 257
 P11 13-MAY-77 13:58 KMC11 ALU TESTS

```

14004 033444 063162 .WORD .S!.DSP
14005 033446 .BZ 55 ;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 .SINC SPAD <2> ;WAS IT 377?
14012 000057 MICPC=MICPC+1
14013 033452 063062 .WORD .S!.DSP
14014 033454 .BC 65 ;YES, NOW LOAD 252.
14015 000060 MICPC=MICPC+1
14016 033454 101064 .WORD .S
14017 033456 .SBR 95 ;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 75 ;
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 15 ;CONTINUE
14039 033474 MICPC=MICPC+1
14040 000070 .WORD .S
14041 033474 100404 .:.*
14042 .:.*
14043 .:.*
14044 .:.*
14045 033476 5S: .MOVE BREG,SPAD <2> ;SET UP FOR CHECKING.
14046 000071 MICPC=MICPC+1
14047 033476 063222 .WORD .S
14048 033500 .MOVE INP1 <CSR0>,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 CNPRE ;VERIFY IT.
14062	033510		MOVE # <MICPC+3>,BREG
14063		000076	MICPC=MICPC+1
14064	033510	000500	.WORD .S.
14065	033512		SBR CNPRE
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 CNPRE ;VERIFY IT.
14072	033516		MOVE # <MICPC+3>,BREG
14073		000101	MICPC=MICPC+1
14074	033516	000503	.WORD .S.
14075	033520		SBR CNPRE
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 CNPRE ;VERIFY IT.
14082	033524		MOVE # <MICPC+3>,BREG
14083		000104	MICPC=MICPC+1
14084	033524	000506	.WORD .S.
14085	033526		SBR CNPRE
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 CNPRE ;VERIFY IT.
14092	033532		MOVE # <MICPC+3>,BREG
14093		000107	MICPC=MICPC+1
14094	033532	000511	.WORD .S.
14095	033534		SBR CNPRE
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 CNPRE ;VERIFY IT.
14102	033540		MOVE # <MICPC+3>,BREG
14103		000112	MICPC=MICPC+1
14104	033540	000514	.WORD .S.
14105	033542		SBR CNPRE
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 CNPRE ;VERIFY IT.
14112	033546		MOVE # <MICPC+3>,BREG
14113		000115	MICPC=MICPC+1
14114	033546	000517	.WORD .S.
14115	033550		SBR CNPRE

```

14116 000116
14117 033550 100650
14118
14119
14120 033552
14121 000117
14122 033552 023001
14123 033553
14124 033554
14125 000120
14126 033554 000522
14127 033556
14128 000121
14129 033556 100650
14130 033560
14131 000122
14132 033560 023021
14133 033562
14134 033562
14135 000123
14136 033562 000525
14137 033564
14138 000124
14139 033564 100650
14140 033566
14141 000125
14142 033566 023041
14143 033570
14144 033570
14145 000126
14146 033570 000530
14147 033572
14148 000127
14149 033572 100650
14150 033574
14151 000130
14152 033574 023061
14153 033576
14154 033576
14155 000131
14156 033576 000533
14157 033600
14158 000132
14159 033600 100650
14160 033602
14161 000133
14162 033602 023101
14163 033604
14164 033604
14165 000134
14166 033604 000536
14167 033606
14168 000135
14169 033606 100650
14170 033610
14171 000136
    
```

```

MICPC=MICPC+1
.WORD .S.
;:
;:
MOVE INPO <0>,SPAD <1> ;GET IBUS REG.
MICPC=MICPC+1
.WORD .S.
CALL CNPRE ;VERIFY IT.
MOVE # <MICPC+3>,BREG
MICPC=MICPC+1
.WORD .S.
SBR CNPRE
MICPC=MICPC+1
.WORD .S.
MOVE INPO <1>,SPAD <1> ;GET NEXT IBUS-REGISTER.
MICPC=MICPC+1
.WORD .S.
CALL CNPRE ;VERIFY IT.
MOVE # <MICPC+3>,BREG
MICPC=MICPC+1
.WORD .S.
SBR CNPRE
MICPC=MICPC+1
.WORD .S.
MOVE INPO <2>,SPAD <1> ;GET NEXT IBUS+REG.
MICPC=MICPC+1
.WORD .S.
CALL CNPRE ;VERIFY IT.
MOVE # <MICPC+3>,BREG
MICPC=MICPC+1
.WORD .S.
SBR CNPRE
MICPC=MICPC+1
.WORD .S.
MOVE INPO <3>,SPAD <1> ;GET NEXT IBUS+REG.
MICPC=MICPC+1
.WORD .S.
CALL CNPRE ;VERIFY IT.
MOVE # <MICPC+3>,BREG
MICPC=MICPC+1
.WORD .S.
SBR CNPRE
MICPC=MICPC+1
.WORD .S.
MOVE INPO <4>,SPAD <1> ;GET NEXT IBUS+REG.
MICPC=MICPC+1
.WORD .S.
CALL CNPRE ;VERIFY IT.
MOVE # <MICPC+3>,BREG
MICPC=MICPC+1
.WORD .S.
SBR CNPRE
MICPC=MICPC+1
.WORD .S.
MOVE INPO <5>,SPAD <1> ;GET NEXT IBU
MICPC=MICPC+1
    
```

14172	033610	023121	.WORD .S.
14173	033612		CALL CMPRE ;VERIFY IT.
14174	033612		MOVE B <MICPC+3>,BREG
14175		000137	MICPC=MICPC+1
14176	033612	000541	.WORD .S.
14177	033614		SBR CMPRE
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 CMPRE ;VERIFY IT
14184	033620		MOVE B <MICPC+3>,BREG
14185		000142	MICPC=MICPC+1
14186	033620	000544	.WORD .S.
14187	033622		SBR CMPRE
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 CMPRE ;VERIFY IT.
14194	033626		MOVE B <MICPC+3>,BREG
14195		000145	MICPC=MICPC+1
14196	033626	000547	.WORD .S.
14197	033630		SBR CMPRE
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 CMPRE ;VERIFY IT.
14211		000151	MOVE B <MICPC+3>,BREG
14212	033636	000553	MICPC=MICPC+1
14213	033640		.WORD .S.
14214		000152	SBR CMPRE
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.

DZKCA MACY11 27(1006) 13-MAY-77 14:07 PAGE 262
 DZKCA.P11 13-MAY-77 13:58 KMC11 ALU TESTS

PAGE

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          MICPC=MICPC+1
14344 033756    010000      .WORD     .S.
14345 033760          MOVE      # 0,MPR          ;
14346          MICPC=MICPC+1
14347 033760    004000      .WORD     .S.
14348 033762          MOVE      BREG,SPAD <0>      ;SET THE COUNTER.
14349          MICPC=MICPC+1
14350 033762    063220      .WORD     .S.
14351 033764          MOVE      # 3,SPAD <3>      ;SET THE COUNTER.
14352          MICPC=MICPC+1
14353 033764    003003      .WORD     .S.
14354 033766          MOVE      MEH,BREG MARINC ;GET THE CONTENTS OF MEMORY LOCATION.
14355          MICPC=MICPC+1
14356 033766    054620      .WORD     .S.
14357 033770          $IFEQ   BREG,SPAD <2> 15$      ;BRANCH IF GOOD.
14358
14359
14360 033770          SUB2C   SPAD <2>,BREG,NOP
14361          MICPC=MICPC+1
14362 033770    060362      .WORD     .S.
14363 033772          BZ      15$
14364          MICPC=MICPC+1
14365 033772    101637      .WORD     .S.
14366 033774          MOVE      MEH,OUT1 <CSR4> ;GOOD DATA
14367          MICPC=MICPC+1
14368 033774    041224      .WORD     .S.
14369 033776          MOVE      BREG,OUT1 <CSR5>      ;BAD DATA
14370          MICPC=MICPC+1
14371 033776    061225      .WORD     .S.
14372 034000          MOVE      # 20,BREG          ;
14373          MICPC=MICPC+1
14374 034000    000420      .WORD     .S.
14375 034002          MOVE      BREG,OUT1 <CSR3>      ;TYPE OF ERROR.
14376          MICPC=MICPC+1
14377 034002    061223      .WORD     .S.
14378 034004          CALL    ERROR1 ;DATA ERROR.
14379 034004          MOVE      # <MICPC+3>,BREG
14380          MICPC=MICPC+1
14381 034004    000234      .WORD     .S.
14382 034006          SBR     ERROR1
14383          MICPC=MICPC+1
14384 034006    104401      .WORD     .S.
14385 034010          SBR     15$ ;LOOP ON ERROR.
14386          MICPC=MICPC+1
14387 034010    100400      .WORD     .S.
14388 034012          $INC   SPAD <0> ;COUNT BY ONE.
14389          MICPC=MICPC+1
14390 034012    063060      .WORD     .S.!.DSP
14391 034014          BC      16$ ;COUNT BY ONE.
14392          MICPC=MICPC+1
14393 034014    101242      .WORD     .S.
14394 034016          SBR     12$ ;DO THE NEXT.
14395          MICPC=MICPC+1
    
```

14396	034016	100625		
14397	034020		168:	WORD \$ SDEC SPAD (3) ;IS IT DONE?? NICPC=NICPC+1
14398		000242		
14399	034020	063163	.WORD	\$!.DSP BZ 178 ;YES.SCOPE IT. NICPC=NICPC+1
14400	034022			
14401		000243		
14402	034022	101645		
14403	034024			
14404		000244		
14405	034024	100625		
14406	034026		178:	WORD \$ CALL SCOPE ;SCOPE THE TEST... MOVE # (NICPC+3),BREG NICPC=NICPC+1
14407	034026			
14408		000245		
14409	034026	000647		
14410	034030			
14411		000246		
14412	034030	104454		
14413	034032			
14414		000247		
14415	034032	100400		
14416	034034		CMPE:	WORD \$ MOVE BREG SPAD (17) ;SAVE THE RETURN ADDRESS. NICPC=NICPC+1
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		

KMC11 ALU TESTS

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 000000
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

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

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

\$XZ
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
JSR PC,LDRWMT ;LOAD-VERIFY-WAIT.
MCT64: MCT64 EKROR 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.
MOVE #0,MLR ; CLEAR MAR.
MICPC=MICPC+1
.WORD .S.
MOVE #0,MPR ; CLEAR MAR.
MICPC=MICPC+1
.WORD .S.
MOVE BREG,OUT1 (CSR4) ; CLEAR BSEL4.
MICPC=MICPC+1
.WORD .S.
MOVE BREG,OUT1 (CSRS) ; CLEAR BSEL5.
MICPC=MICPC+1
.WORD .S.
MOVE BREG,OUT0 (0) ; CLEAR IN DATA LB.
MICPC=MICPC+1
.WORD .S.
MOVE BREG,OUT0 (1) ; CLEAR IN DATA HB.
MICPC=MICPC+1
.WORD .S.
MOVE BREG,OUT0 (4) ; CLEAR IN BA LB.
MICPC=MICPC+1
.WORD .S.
MOVE BREG,OUT0 (5) ; CLEAR IN BA HB.
MICPC=MICPC+1
.WORD .S.
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,OUT0 <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,OUT0 <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	034150		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.	

```

000034      MOVE      BREG,OUT1 <CSR10>      ;
061230      MICPC=MICPC+1
              .WORD      .S.
000035      55:      MOVE      INP1 <CSR10>,BREG      ;IS NPR DONE?
120600      MICPC=MICPC+1
              .WORD      .S.
              BBO      65
000036      MICPC=MICPC+1
122044      55:      MOVE      INP1 <CSR2>,BREG      ;IS CSR LOADED???
              .WORD      .S.
              BBI      45
              MICPC=MICPC+1
              .WORD      .S.
              MICPC=MICPC+1
122400      .WORD      .S.
14578      034220      MOVE      INP1 <CSR4>,SPAD <1>      ;
14579      034222      MICPC=MICPC+1
14580      .WORD      .S.
14581      034222      123101      MOVE      INP1 <CSR5>,SPAD <2>      ;
14582      034224      MICPC=MICPC+1
14583      .WORD      .S.
14584      034224      123122      SBR      125      ;GO CHECK DATA...
14585      034226      MICPC=MICPC+1
14586      .WORD      .S.
14587      034226      100451      55:      MOVE      INP1 <CSR2>,BREG      ;IS CSR LOADED???
14588      034230      MICPC=MICPC+1
14589      .WORD      .S.
14590      034230      120440      BBI      55      ;NO,CHECK NPR DONE...
14591      034232      MICPC=MICPC+1
14592      .WORD      .S.
14593      034232      102435      MOVE      INP1 <CSR4>,SPAD <1>      ;TRANSFER TO SPAD 1.
14594      034234      MICPC=MICPC+1
14595      .WORD      .S.
14596      034234      123101      MOVE      INP1 <CSR5>,SPAD <2>      ;TRANSFER TO SPAD 2.
14597      034236      MICPC=MICPC+1
14598      .WORD      .S.
14599      034236      123122      SBR      55      ;GO CHECK NPR.
14600      034240      MICPC=MICPC+1
14601      .WORD      .S.
14602      034240      100435      125:      MOVE      INPD <0>,SPAD <0>      ;GET NPR DATA.
14603      034242      MICPC=MICPC+1
14604      .WORD      .S.
14605      034242      023000      MOVE      SPAD <1>,BREG      ;GET CSR DATA.
14606      034244      MICPC=MICPC+1
14607      .WORD      .S.
14608      034244      060601      SIFNE      BREG,SPAD <1> 155      ;BR IF NO COMPARE.
14609      034246
14610
14611
14612      034246      SUB2C      SPAD <1>,BREG,NOP
14613      .WORD      .S.
14614      034246      060361      MICPC=MICPC+1
14615      034250      .WORD      .S.
14616      .WORD      .S.
14617      034250      101456      BZ      645
14618      034252      MICPC=MICPC+1
14619      .WORD      .S.
              SBR      155
              MICPC=MICPC+1

```

14620	034252	100462	.WORD .S.	
14621	034254		645: MOVE INPO <1>,SPAD <0>	;GET NPR DATA.
14622	034254		MICPC=MICPC+1	
14623		000056	.WORD .S.	
14624	034254	023020	MOVE SPAD <2>,BREG	;GET CSR DATA.
14625	034256		MICPC=MICPC+1	
14626		000057	.WORD .S.	
14627	034256	060602	\$IFEQ BREG,SPAD <2> 165	;BRANCH IF GOOD.
14628	034260			
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 # 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 # <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 IS	;LOOP ON ERROR.
14660		000071	MICPC=MICPC+1	
14661	034302	100400	.WORD .S.	
14662	034304		165: CALL SCPE	;SCOPE THE TEST.
14663	034304		MOVE # <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 IS	;YES DO NEXT PASS.
14670		000074	MICPC=MICPC+1	
14671	034310	100400	.WORD .S.	
14672	034312	125125	HEL DAT: 125125	
14673	034314		SPWR1	
14674	034314		SXZ	
14675				

```

14676
14677 ;***** TEST 65 *****
14678 ;*FORCE POWER FAIL TEST.
14679 ;*SET FORCE POWER FAIL VERIFY THAT PROCESSOR TRAPS TO LOC 24.
14680 ;*GOING DOWN AND COMING UP. VERIFY ALSO THAT BUS INIT WAS
14681 ;*BLOCKED FROM GETTING TO KMC DURING THE POWER FAIL .
14682 034314 SXZ ;*****
14683
14684
14685
14686 034314 STSTN
14687 ; TEST 65
14688 ;-----
14689 034314 012737 000065 001202 TST65: MOV #65,STSTNM ; LOAD THE NO. OF THIS TEST
14690 034322 012737 003662 001442 MOV #SEOP,NEXT ; POINT TO THE END OF PASS HANDLER.
14691 ;R1 CONTAINS BASE KMC11 ADDRESS
14692 034330 004737 035536 JSR PC,LDRVMT ;LOAD-VERIFY-WAIT.
14693 034334 034350 MCT65 ERROR 22 ;TIME OUT ERROR...
14694 034336 104022 MOV #STACK,SP ;RESET STACK...
14695 034340 012706 001200 JMP @NEXT ;GO TO NEXT TEST...
14696 034344 000177 145072
14697 034350 MCT65:
14698 034350 15: MOVE #0,BREG ;
14699 000000 NICPC=NICPC+1 ;
14700 034350 000400 .WORD .S. ;
14701 034352 MOVE BREG,SPAD <16> ;FOR RETURN ADDRESS ...
14702 000001 NICPC=NICPC+1 ;
14703 034352 063236 .WORD .S. ;
14704 034354 MOVE BREG,SPAD <1> ;
14705 000002 NICPC=NICPC+1 ;
14706 034354 063221 .WORD .S. ;
14707 034356 MOVE BREG,SPAD <2> ;
14708 000003 NICPC=NICPC+1 ;
14709 034356 063222 .WORD .S. ;
14710 034360 MOVE BREG,SPAD <3> ;
14711 000004 NICPC=NICPC+1 ;
14712 034360 063223 .WORD .S. ;
14713 034362 MOVE #220,BREG ;GET READY TO SETUP POWER FAIL..
14714 000005 NICPC=NICPC+1 ;
14715 034362 000620 .WORD .S. ;
14716 034364 MOVE INP1 <CSRO>,SPAD <0> ;GET CNTL/O REGISTER..
14717 000006 NICPC=NICPC+1 ;
14718 034364 123000 OR BREG,SPAD <0>,OUT1 <CSRO> ;SET RDO, POWER FAIL BIT..
14719 034366 000007 NICPC=NICPC+1 ;
14720 000007 .WORD .S. ;
14721 034366 061300 MOVE #200,BREG ;
14722 034370 000010 NICPC=NICPC+1 ;
14723 000010 .WORD .S. ;
14724 034370 000600 MOVE INP1 <CSR11>,SPAD <0> ;GET UPMS REGISTER.
14725 034372 000011 NICPC=NICPC+1 ;
14726 000011 .WORD .S. ;
14727 034372 123220 OR BREG,SPAD <0>,OUT1 <CSR11> ;SET BR+REQ,VCTR:=XXD..
14728 034374 000012 NICPC=NICPC+1 ;
14729 000012 .WORD .S. ;
14730 034374 061311 25: MOVE INP1 <CSR11>,BREG ;IS REQ GRANTED??
14731 034376

```

```

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 # 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          MOVE INP1 <CSRD>,BREG ;IS POWER FAIL SERVED??
14757 034414          MICPC=MICPC+1
14758          000022          .WORD .S.
14759 034414 120400          BB7 65 ;NO, UPDATE DELAY COUNT.
14760 034416          MICPC=MICPC+1
14761          000023          .WORD .S.
14762 034416 103425          SBR 95 ;YES, SCOPE THE TEST..
14763 034420          MICPC=MICPC+1
14764          000024          .WORD .S.
14765 034420 100432          65: $INC SPAD <1> ;UPDATE DELAY COUNT.
14766 034422          MICPC=MICPC+1
14767          000025          .WORD $!.DSP
14768 034422 063061          $ADC SPAD <2> ;UPDATE DELAY COUNT.
14769 034424          MICPC=MICPC+1
14770          000026          .WORD $!.DSP
14771 034424 063102          $ADC SPAD <3> ;UPDATE DELAY COUNT.
14772 034426          MICPC=MICPC+1
14773          000027          .WORD $!.DSP
14774 034426 063103          BC 125 ;IS IT DELAYED ENOUGH?
14775 034430          MICPC=MICPC+1
14776          000030          .WORD .S.
14777 034430 101032          SBR 45 ;NO, WAIT
14778 034432          MICPC=MICPC+1
14779          000031          .WORD .S.
14780 034432 100422          125:
14781 034434          95: MOVE INP1 <CSRD>,BREG ;DUMMY WAIT LOOP
14782 034434          MICPC=MICPC+1
14783 034434          .WORD .S.
14784          000032          SBR 95 ;DO THE NEXT PASS.
14785 034434 120400          MICPC=MICPC+1
14786 034436          .WORD .S.
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 XXX.
* 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
ERROR: SINC   SPAD (16)      ;PREPARE FOR RETURN.
        MICPC=MICPC+1
        .WORD .S.!.DSP
ERROR1: SINC   SPAD (16)
        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 XXX.
        MICPC=MICPC+1
        .WORD .S.

```

```

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

```

```

51$: MOVE INP1 <CSR11>,BREG ;IS BR+RO GRANTED?
    MICPC=MICPC+1
    .WORD $.
    BBT $1$ ;NO, SPIN ON IT
    MICPC=MICPC+1
    .WORD $.
52$: MOVE INP1 <CSRD>,BREG ;IS ERROR INTERRUPT SERVICED?
    MICPC=MICPC+1
    .WORD $.
    BBO $2$ ;NO, SPIN ON IT.
    MICPC=MICPC+1
    .WORD $.
    BBT $3$ ;LOOP ON ERROR?
    MICPC=MICPC+1
    .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 $.
53$: MOVE $ 0,BREG ;
    MICPC=MICPC+1
    .WORD $.
    MOVE BREG SPAD <16> ;RESET SPAD <16>...
    MICPC=MICPC+1
    .WORD $.
    MOVE BREG OUT1 <CSRD> ;RESET THE TALK REGISTER...
    MICPC=MICPC+1
    .WORD $.
    SBR SPAD <17> PAGED ;RETURN TO CALLER.
    MICPC=MICPC+1
    .WORD $.

```

```

*****
*
* SCOPE DATA ROUTINE
*
*****

```

```

SCP10: $INC SPAD <16> ;PREPARE FOR RETURN.
    MICPC=MICPC+1
    .WORD $.!-DSP
SCP110: $INC 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 <CSRD> ;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+RO AT XX0.
14912      000440      MICPC=MICPC+1
14913      034540      061311      .WORD      .S.
14914      034542      545: MOVE      INP1 <CSR11>,BREG      ;IS BR+RO GRANTED?
14915      000441      MICPC=MICPC+1
14916      034542      120620      .WORD      .S.
14917      034544      BB7      545      ;NO, SPIN ON IT.
14918      000442      MICPC=MICPC+1
14919      034544      107441      .WORD      .S.
14920      034546      555: MOVE      INP1 <CSRD>,BREG      ;IS SCOPI SERVICED?
14921      000443      MICPC=MICPC+1
14922      034546      120400      .WORD      .S.
14923      034550      BB1      555      ;NO, SPIN ON IT.
14924      000444      MICPC=MICPC+1
14925      034550      106443      .WORD      .S.
14926      034552      BB7      555      ;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      565: 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      * SCOPE THE TEST ROUTINE.
14951      *
14952      .....
14953
14954      034570      SCPE0: MOVE      BREG,SPAD <17> ;SAVE RETURN ADDRESS.
14955      000454      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 034630 LPCSQR: 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          MICPC=MICPC+1
15014 034632 000475   .WORD .S
15015 034634 061223   MOVE #201,BREG                ;SET UP FOR ERROR REPORT.
15016          MICPC=MICPC+1
15017 034634 000476   .WORD .S
15018 034636 000601   MOVE BREG,OUT1 (CSR0)          ;
15019          MICPC=MICPC+1
15020 034636 061220   .WORD .S
15021 034640          MOVE #200,BREG                ;BREG+BR-R0. AT VCTR:=XX0.
15022          MICPC=MICPC+1
15023 034640 000500   .WORD .S
15024 034642          MOVE BREG,CJT1 (CSR11)           ;IS INTERRUPT SERVICED??
15025          MICPC=MICPC+1
15026 034642 000501   .WORD .S
15027 034644 061231   MOVE INP1 (CSR11),BREG        ;IS INTERRUPT SERVICED??
15028          MICPC=MICPC+1
15029 034644 120620   .WORD .S
15030 034646          BR7 15
15031          MICPC=MICPC+1
15032          .WORD .S
15033 034646 000503   MOVE INP1 (CSR0),BREG        ;IS ERROR SERVICED??
15034          MICPC=MICPC+1
15035 034650 120400   .WORD .S
15036 034652          BR0 25
15037          MICPC=MICPC+1
15038 034652 000505   .WORD .S
15039 034654          CALL SCPE0
15040 034654          MOVE # (MICPC+3),BREG           ;SCOPE THE FAILING TEST.
15041          MICPC=MICPC+1
15042 034654 000506   .WORD .S
15043 034656          BR SCPE0
15044          MICPC=MICPC+1
15045 034656 000507   .WORD .S
15046 034660          MOVE INP1 (CSR0),BREG        ;WAIT FOR PDP11 TO
15047          MICPC=MICPC+1
15048 034660 120400   .WORD .S
15049 034662          BR 35
15050          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.
:
*****

```

```

15053 034664          INISR: .MOV R0,-(SP)          ;PUSH R0.
15054 034664 013737 001442 035356 .MOV NEXT,SVIDT          ;SAVE NEXT TEST ADDRESS.
15055 034672 013737 001206 035354 .MOV SLPADR,SVLPDR       ;SAVE LOOP ADDRESS.
15056 034700 032711 000001 .BIT #BIT0,(R1)          ;IS ERROR BIT SET?
15057          BNE ERR1          ;YES, SERVE THE ERROR INTERRUPT.
15058 034704 001012          BIT #BIT1,(R1)          ;IS IT SCOPE DATA SET?
15059          .WORD .S
15060          .WORD .S
15061          .WORD .S
15062          .WORD .S
15063          .WORD .S
15064          .WORD .S
15065          .WORD .S
15066          .WORD .S
15067          .WORD .S

```

15068	034712	001075			BNE	SCPDT			YES, GO SERVE IT.
15069	034714	032711	000020		BIT	#BIT4, (R1)			IS PWR. FAIL SERVICE SET?
15070	034720	001106			BNE	PMFAIL			GO SERVE POWER FAIL INTERRUPT.
15071									
15072	034722	104401	011032		TYPE	,MILINT			TYPE ILLEGAL INTERRUPT.
15073	034726	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		MOVW	4(R1), SCNDAT		LOAD GOOD DATA.
15078	034734	116137	000005	001226		MOVW	5(R1),		LOAD BAD DATA.
15079	034736	116137	000006	001222		MOVW	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), ES		SET THAT IN ERROR TRAP.
15082	035004	116137	000007	001220		MOVW	7(R1), SCNDOR		SET CSR FOR REFERENCE.
15083	035012	012737	035354	007070		MOV	#SVLADR, ERTAB1		SAVE ERROR PC...
15084									
15085	035020	104000			6S:	ERROR			REPORT THE ERROR.
15086	035022	012737	001460	007070	3S:	MOV	#SAVPC, ERTAB1		RESTORE ERROR PC POINTER.
15087	035030	032777	000400	144202		BIT	#SW08, JSWR		GO TO TOP OF TEST?
15088	035036	001020				BNE	IS		BRANCH IF YES.
15089	035040	032777	002000	144172		BIT	#SW10, JSWR		GO TO NEXT TEST?
15090	035046	001412				BEQ	2S		BRANCH IF NO.
15091	035050	013737	035356	001206		MOV	SVNXT, SLPADR		SET FOR NEXT TEST.
15092	035056	012706	001200			MOV	#STACK, SP		RESET SP.
15093	035062	012737	000200	177776		MOV	#PR4, PS		SET PRIORITY 4 SO KMC CAN INTRRUPT.
15094	035070	000177	144112			JMP	2SLPADR		GO TO SPECIFIED TEST.
15095	035074	042711	000200		2S:	BIC	#BIT7, (R1)		CLEAR LOOP ON ERROR BIT.
15096	035100				1S:	MOV	#1, CLKCTR		RESET CLOCK.
15097	035100	042711	000001			BIC	#BIT0, (R1)		CLEAR ERROR BIT IN BSELO.
15098	035104	000511				BR	RETN1		RETURN.
15099	035106	005037	001444		SCPDT:	CLR	LOCK		CLEAR FLAG
15100	035112	104405				SCOPI			SCOPE THE DATA.
15101	035114	005737	001444			TST	LOCK		IS FLAG SET?-I.E SCOPE THE DATA.
15102	035120	001002				BNE	SCPDT1		YES, GO SET ACCORDINGLY.
15103	035122	042711	000200			BIC	#BIT7, (R1)		CLEAR SCOPE DATA LOOP BIT.
15104	035126				SCPDT1:				
15105	035126	042711	000002			BIC	#BIT1, (R1)		CLEAR SCOPE DATA BIT.
15106	035132	000476				BR	RETN1		RETURN.
15107					***				THIS IS PART OF POWER FAIL TEST
15108					***				AFTER KMC CAUSES FORCE POWER FAIL
15109					***				AND SETTING UP FOR POWER FAIL THIS
15110					***				FUNCTIONS DONE HERE.
15111	035134	012737	035202	001442	PMFAIL:	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	#12, #24		SET UP FOR FORCE POWER FAIL.
15116	035166	042711	000020			BIC	#BIT4, (R1)		SIGNAL POWER FAIL SET UP.
15117	035172	005237	011112		15S:	INC	TEMP		WAIT FOR POWER FAIL.
15118	035176	001375				BNE	15S		BR IF DELAY NOT DONE.
15119	035200	104017				ERROR	17		ERROR, NO POWER FAIL.
15120	035202	000427			30S:	BR	33S		
15121					***				
15122					***				RESTARTS HERE IN CASE OF POWER FAIL.
15123					***				

```

15124 035204 012737 035222 000024 128:  MOV    $215,2824    ; POWER UP ADDRESS.
15125 035212 010637 035220                MOV    SP,248      ; STORE STACK.
15126 035216 000000                HALT                    ; WAIT FOR POWER UP SEQUENCE.
15127 035220 000000                WORD    0
15128 035220 013706 035220 248:  MOV    248,SP
15129 035220 000000                POP2SP                ; RESTORE STACK.
15130 035220 012737 035262 001206 218:  MOV    $338,SLPADR   ; POP STACK THICE.
15131 035220 012737 035262 001442                MOV    $338,NEXT
15132 035220 012737 000024                MOV    (SP),2824
15133 035220 012737 007100 000024                CMP    $PADR0N,2824
15134 035220 001401                BEQ    16
15135 035220 104017                ERROR 17
15136 035220 012737 007100 000024 338:  MOV    $PADR0N,2824
15137 035220 012706 001200                MOV    $STACK,SP
15138 035220 005711                TST    (R1)
15139 035220 100407                BHI    RETN2
15140 035220 012737 035316 001206                MOV    $RETN2,SLPADR
15141 035220 012737 035316 001442                MOV    $RETN2,NEXT
15142 035220 104016                ERROR 16
15143 035316 013737 035356 001206 RETN2: MOV    SVXKT,SLPADR
15144 035316 000177 143656                JMP    $SLPADR
15145 035330 012737 000001 036150 RETN1: MOV    $1,CLKCTR
15146 035330 013737 035356 001442                MOV    SVXKT,NEXT
15147 035330 013737 035354 001206                MOV    SVLPDR,SLPADR
15148 035330 000002                RTI
15149 035330 000000                SVLPDR: 0
15150 035356 000000                SVXKT: 0

```

```

*****
*
*   OUTPUT INTERRUPT SERVICE ROUTINE
*   1. TAKE CARE OF SCOPE TEST
*
*****

```

```

15162 035360 132761 000020 000002 OUISR: BITB    $BIT4,2(R1)    ; IS IT FOR HELL RAISER??
15163 035366 001015                BNE    HERISR          ; YES DO THAT.
15164 035370 013737 001206 035534                MOV    $SLPADR,$SLPADR ; SAVE LOOP ADDRESS.
15165 035376 012737 035452 001206                MOV    $SCOPE,$SLPADR ; STORE RETURN ADDRESS.
15166 035404 000004                SCOPE                ; SCOPE THE TEST.
15167 035406 013737 035534 001206                MOV    $SVLPDR,$SLPADR ; NO ITERATION.
15168 035414 005137 036146                COM    DONE
15169 035420 000432                BR     RET1            ; SETUP DONE FLAG
15170 035422 142761 000020 000002 HERISR: BICB    $BIT4,2(R1)    ; YES ALL SET FOR HELL RAISER.
15171 035430 013761 034312 000004                MOV    $HELDAT,4(R1)  ; LOAD CSR.
15172 035436 142761 000002 000002                BICB    $BIT1,2(R1)   ; CLEAR LOAD CSR BIT.
15173 035444 105061 000002                CLAB   2(R1)          ; CLEAR CSR2.
15174 035450 000416                BR     RET1
15175 035452 013737 035534 001206 SKOPE: MOV    $SVLPDR,$SLPADR ; RESTORE RETURN ADDRESS.
15176                BIC    $BIT7,2(R1)    ; CLEAR RD=0.
15177 035460 122761 000024 000003                CMPB   $24,3(R1)     ; IS IT UPC $0. ERROR??
15178 035466 001013                BNE    RET2           ; NO THEN LET IT BE SCOPED BY
15179                ; MICRO-PROCESSOR SCOPE ROUTINE.

```

```

15180 035470 105061 000003 CLR 3(R1) ; CLEAN THE WORLD.
15181 035474 013737 001206 001442 MOV SVLPAD, NEXT ; ELSE, START FROM BEGINING.
15182 035502 006137 036156 COM DONE
15183 035506 012737 000001 036150 RET1: MOV R1, CLKCTR ; RESET CLOCK.
15184 035514 000002 RTI ; RETURN TO WAIT CLOCK.
15185 035516 012737 000001 036150 RET2: MOV R1, 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
15201
15202
15203
15204 035536 LDVMT:
15205 035536 MCRAM: ; CLR RD ; RD-POINTS TO CRAM ADDRESS.
15206 035536 012700 002000 MOV #2000, RD ; MASTER CLEAR KMC11
15207 035536 012711 040000 MOV #BIT14, (R1) ; AND SHUT IT OFF.
15208 035536 042711 140000 BIC #BIT15!BIT14, (R1) ; RE-SET UP CSR (MICRO-INSTRUCTION)
15209 035536 012702 104474 MOV #104474, R2 ; START WITH CLEAN WORLD.
15210 035536 005011 65: CLR (R1) ; START WITH CLEAN WORLD.
15211 035536 010061 000004 MOV RD, 4(R1) ; LOAD CRAM ADDRESS.
15212 035536 010261 000006 MOV R2, 6(R1) ; LOAD INSTRUCTION.
15213 035536 012711 002000 MOV #BIT10, (R1) ; SET RD+0.
15214 035536 012711 022000 MOV #BIT13!BIT10, (R1) ; WRITE IT.
15215 035536 005300 DEC RD ; COUNT BY ONE.
15216 035536 001365 BNE 65 ; BR IF NOT DONE.
15217 035536 013602 MOV #2(SP)+R2 ; SAVE START ADDRESS...
15218 035536 013237 036152 MOV R2, TSTR ; POP RETURN ADDRESS...
15219 035536 012711 000002 ADD #2, -(SP)
15220 035536 013737 001442 036154 MOV NEXT, MCEND ; SET UP END POINTER.
15221 035536 022737 003662 036154 CMP #NEOP, MCEND ; IS IT LAST TEST?
15222 035536 010003 15: BNE 15 ; NO, THEN IT'S O.K.
15223 035536 012737 034440 036154 MOV #STSEN, MCEND ; SET UP END POINTER.
15224 035536 005011 15: CLR (R1) ; START WITH CLEAN WORLD.
15225 035536 010061 000004 MOV RD, 4(R1) ; LOAD CRAM ADDRESS.
15226 035536 012261 000006 MOV (R2)+, 6(R1) ; LOAD WORD TO BE WRITTEN.
15227 035536 012711 002000 MOV #BIT10, (R1) ; WRITE IT...
15228 035536 012711 022000 MOV #BIT13!BIT10, (R1) ; WRITE IT.
15229 035536 005200 INC RD ; UPDATE CRAM ADDRESS.
15230 035536 022700 002000 CMP #2000, RD ; OVER FLOW?
15231 035536 002403 15: BLT 25 ; YES, THATS IT! NO MORE.
15232 035536 023702 036154 CMP MCEND, R2 ; IS IT DONE?
15233 035700 003360 BGT 15 ; NO, CONTINUE LOADING.
15234 035702 022737 034664 036154 25: CMP #MCSREN, MCEND ; IS UTILITY ALSO LOADED?
15235 035710 001410 BEQ 35 ; YES, RETURN.

```


KMC11 ALU TESTS

```

1 036076 012711 100000      MOV      #BIT15, (R1)      ; INITIATE THE TEST...
1 036102 005737 036146      15:     TST      DONE           ; IS DONE FLAG SET?
1 036106 001010          BNE     38              ; YES, START THE NEXT TEST.
1 036110 005237          INC     CLKCTR         ; COUNT BY ONE.
1 036114 005337          DEC     CLKCTR         ; WAIT CLOCK.
1 036120 005237          INC     CLKCTR         ;
1 036126 001388          BNE     18              ; GO CHECK AGAIN. IF NOT TIMED OUT.
1 036130 000207          RTS     PC              ;
1 036136 013737          MOV     NEXT_SLPADR    ; SET THE NEXT TEST ADDRESS.
1 036138 012706          MOV     #STACK_SP     ; POP THE STACK.
1 036144 000177          JMP     @SLPADR        ; START THE NEXT TEST.
1 036146 000000      DONE:    0
1 036150 000000      CLKCTR:  0
1 036154 000000      TSTRT:   0
1 036158 000000      MCEND:   0
15 036164 043516 040517 044504  MLCR:    .ASCIZ  <200>/LOADING ERROR/
15 036172 051117 042440 051122
15 036176 036176      .EVEN
15 036176 044600 052502 027523  EMI1:    .ASCIZ  <200>/IBUS/IBUS# REGISTER DATA TEST/
15 036235 200 041111 051525  EMI2:    .ASCIZ  <200>/IBUS/IBUS# REGISTER DUAL ADDRESSING TEST/
15 036307 200 051102 051040  EMI3:    .ASCIZ  <200>/NPR REGISTER DATA TEST/
15 036336 051600 051103 052101  EMI4:    .ASCIZ  <200>/SCRATCH PAD DATA TEST/
15 036365 200 041523 040522  EMI5:    .ASCIZ  <200>/SCRATCH PAD DUAL ADDRESSING TEST/
15 036427 200 040515 047111  EMI6:    .ASCIZ  <200>/MAIN MEMORY DATA TEST/
15 036456 046600 044501 020116  EMI7:    .ASCIZ  <200>/MAIN MEMORY DUAL ADDRESSING TEST/
15 036520 040600 052125 020117  EMI10:   .ASCIZ  <200>/AUTO INCRING FUNCTION TEST/
15 036553 200 050116 020122  EMI11:   .ASCIZ  <200>/NPR TEST/
15 036585 200 052515 052114  EMI12:   .ASCIZ  <200>/MULTIPLE NPR TEST/
15 036610 047200 047117 042440  EMI13:   .ASCIZ  <200>/NON EX MEM FAILED TO SET/
15 036642 050200 047522 051107  EMI14:   .ASCIZ  <200>/PROGRAM CLOCK TEST/
15 036666 040600 052514 043040  EMI15:   .ASCIZ  <200>/ALU FUNCTION WITH C BIT CLEAR TEST/
15 036732 050200 053517 051106  EMI16:   .ASCIZ  <200>/POWER FAIL; BUS INIT WAS NOT BLOCKED/
15 037000 042200 051117 042503  EMI17:   .ASCIZ  <200>/FORCE POWER FAIL ERROR/
15 037030 047200 044517 042523  EMI20:   .ASCIZ  <200>/NOISE TEST ON IBUS, IBUS, SPAD, MEMORY/
15 037076 040600 052514 041440  EMI21:   .ASCIZ  <200>/ALU C BIT TEST FAILURE/
15 037126 052200 046511 020105  EMI22:   .ASCIZ  <200>/TIME OUT ERROR/
15 037146 040600 052514 043040  EMI23:   .ASCIZ  <200>/ALU FUNCTION TEST WITH C BIT SET/
15 037210 052500 041520 051440  EMI24:   .ASCIZ  <200>/UPC SEQUENCE ERROR/
15 037234 044200 046105 020114  EMI25:   .ASCIZ  <200>/HELL RAISER TEST/

037256 043600 047517 020104  DH1:     .ASCIZ  <200>/GOOD      BAD      UPC      REGISTER/
037317 200 047507 042117  DH2:     .ASCIZ  <200>/GOOD      BAD/
037334 043600 047517 020104  DH3:     .ASCIZ  <200>/GOOD      BAD      UPC      ADDRESS/
037373 200 047507 042117  DH4:     .ASCIZ  <200>/GOOD      BAD      UPC      FUNCTION/
037434 037434      .EVEN

037434 000004      DT1:    4
037436 003 006      .BYTE  3,6
037440 001224      $GOODAT
037442 003 004      .BYTE  3,4
037444 001226      $BODAT
037446 003 006      .BYTE  3,6
037450 001222      $BODADR
037452 003 006      .BYTE  3,6

```

F08

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

PAGE: 0303

037454	001220			SGDAOR	
037456	000002			2	
037460	003	004		BYTE	3,4
037462	001224			SGDAT	
037464	003	004		BYTE	3,4
037466	001226			SEDAT	
037470	000001			CORMAX:	.END

CRLF = 000200	77#	1135	1174										
CSR	1769#	1980											
CSRMAP	012154	1956#											
CSR0 = 000000	1#	13876	14049	14717	14720	14738	14758	14784	14824	14827	14851	14872	14897
	14921	14942	15019	15034	15047								
CSR1 = 000001	1#												
CSR10 = 000010	1#	7362	7404	7407	7548	7608	7611	7614	7638	7641	7794	7821	7922
	7982	7985	8021	14565	14568								
CSR11 = 000011	1#	7551	7554	7812	7815	7824	7858	7861	13729	13754	13757	13783	14532
	14544	14553	14726	14729	14732	14750	14753	14830	14842	14845	14900	14912	14915
	14964	14976	14979	14988	14991	15025	15028						
CSR2 = 000002	1#	13879	14059	14525	14528	14574	14583	14961	14994				
CSR3 = 000003	1#	2321	2401	2518	2521	2715	2735	2912	2942	3119	3208	3345	3443
	3560	3640	3757	3837	3864	4034	4151	4231	4348	4428	4545	4625	4742
	4822	4939	5019	5133	5203	5319	5404	5526	5611	5733	5818	5940	6025
	6147	6226	6354	6433	6551	6646	6768	6853	6975	7060	7182	7267	7437
	7471	7595	7674	7814	7842	8072	8348	8501	8591	8751	8878	9193	9510
	9827	10144	10461	10778	11095	11412	11729	12046	12363	12680	12997	13314	13631
CSR4 = 000004	13738	13770	13808	13888	14029	14376	14444	14550	15013				
	1#	2312	2342	2399	2399	2706	2785	2903	2983	3110	3199	3336	3434
	3551	3631	3748	3828	3945	4025	4142	4222	4339	4419	4536	4616	4733
	4813	4930	5010	5127	5197	5307	5392	5514	5599	5721	5806	5928	6013
	6135	6220	6342	6427	6549	6634	6756	6841	6963	7048	7170	7255	7431
	7465	7662	7702	8026	8182	8339	8488	8685	8745	9184	9501	9818	10135
	10452	10769	11086	11403	11720	12037	12354	12671	12988	13305	13622	13939	14079
CSR5 = 000005	14367	14438	14580	14580	14580	14580	14580	14580	14580				
	1#	2315	2345	2512	2512	2709	2789	2906	2986	3113	3202	3339	3437
	3554	3634	3751	3831	3948	4028	4145	4225	4342	4422	4539	4619	4736
	4816	4933	5013	5130	5200	5313	5398	5520	5605	5727	5812	5934	6019
	6141	6226	6348	6433	6555	6640	6762	6847	6969	7054	7176	7261	7425
	7462	7668	7708	8032	8188	8342	8489	8685	8739	9187	9504	9821	10138
	10455	10772	11089	11406	11723	12040	12357	12674	12991	13308	13625	13942	14089
CSR6 = 000006	14370	14432	14493	14554	14598								
CSR7 = 000007	1#	13892	14099	14815	14598								
	1#	2327	2407	2524	2604	2721	2801	2918	2998	3125	3214	3351	3449
	3566	3646	3763	3843	3960	4040	4157	4237	4354	4434	4551	4631	4748
	4828	4945	5025	5143	5210	5332	5417	5539	5624	5746	5831	5953	6238
	6360	6445	6567	6652	6774	6859	6981	7066	7188	7273	8197	8354	8507
	9128	9199	9445	9516	9762	9833	10079	10150	10396	10467	10713	10784	11030
	11101	11347	11418	11664	11735	11981	12052	12298	12369	12615	12686	12932	13003
	13249	13320	13566	13637	13895	14109							
CYCLE	011464	948	1009	1861#									
DATABP	006754	1600#	1603		1628#								
DATACL =	104413	1564#											
DATAND	006742	1599#	1621	1624#									
DATIN =	000001	1#											
DATIO =	000000	1#											
DATI1 =	000001	1#											
DATOUT =	000021	1#											
DAT00 =	000002	1#											
DAT01 =	000003	1#											
DOISP =	177570	83#	239#	268									
DELAY =	104411	1562#											
DEVTAB	003342	836	883#										
DH1	037256	418	421	427	430	15312#							
DH2	037317	424	442	445	463	478	15312#						

CROSS REFERENCE TABLE -- USER SYMBOLS

TPVEC =	000064	174#		
TRAPVE =	000034	172#		
TRTVEC =	000014	167#		
TSTEN	034440	14789#	15223	15260
TSTRY	036152	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	7988#	
TST36	023606	7989	8128#	
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	13847#	
TST64	034064	13848	14471#	

CROSS REFERENCE TABLE -- USER SYMBOLS

		8132	8136	8137*	8274	8280	8283	8288	8289	8293	8294*	8434	8441	8444
		8149	8150	8154	8155*	8147	8154	8157	8162	8163	8167	8168*	8113	8120
		8223	8228	8224	8233	8234*	8232	8246	8249	8254	8255	8259*	8260*	8253
		8263	8266	8271	8272	8276	8277*	8270	8280	8283	8289	8294*	8293	8294*
		8287	8287	8290	8295	8296	8296	8296	8296	8296	8296	8296	8296	8296
		10228*	10521	10531	10534	10539	10540	10544	10545*	10838	10848	10851	10856	10857
		10861	10862*	11155	11165	11168	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	13845	13845	13850	13851	13855	13856*	14453	14466	14469	14474	14475	14479*
		14480*	14675	14683	14687	14692	14693	14697	14698*	15312*				
SNUL	001254	273*	1145	1174										
SOVER	004334	1044	1047	1058	1070	1076*								
SPASS	001324	303*	368*	380	992*	993*	1010	1018	1066	1082	1885*			
SPASTH	002042	500*												
SPARDN	007100	211	734	1666*	1701	15133	15136							
SPANG	007264	1704*												
SPARUP	007152	1676	1682*											
SOLES	001312	291*	1174	1312	1329	1386	1389	1409	1938	2010	2024	2038		
SROCHR	005134	1249*	1555											
SRODEC	***** U	1558												
SROLN	005254	1277*	1556											
SROCT	005554	1350*	1557											
SROSZ	000007	1270*												
SREGD	001260	277*												
SREGO	001262	279*	1445*	1450										
SREG1	001264	280*	868*	880	1444*	1451								
SREG2	001266	281*	1443*	1452										
SREG3	001270	282*	1442*	1453										
SREG4	001272	283*	1441*	1454										
SREG5	001274	284*	1440*	1455										
SRTNAD	004102	1009*												
SR2A	= ***** U	1558												
SS	= 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*										
SSAVRE	= ***** U	1558												
SSAVR6	007274	1675*	1683	1684*	1685*	1708*								
SSCOPE	004134	209	1038*	2190										
SSETUP	= 000000	991	1039	1238	1335									
SSVLAD	004316	1055	1073*											
SSVPC	= 000040	221*	226											
SSHR	= 164000	1*	42	290	291	963	991	1002	1008	1010	1032	1033	1034	1035
		1046	1058	1060	1061	1062	1063	1064	1076	1081	1705			
SSAREG	001340	311*	752											
SSAPAK	= 000000	1035												
STESTN	001322	302*	1074*											
STRES	001210	290*	991*	1063*	1069	1072*	1081							

CROSS REFERENCE TABLE -- USER SYMBOLS

5344	5349	5354	5359	5364	5369	5374	5379	5384	5389	5394	5399	5404	5409	5414	5419	5424	5429	5434	5439	5444	5449	5454	5459	5464	5469	5474	5479	5484	5489	5494	5499	5504	5509	5514	5519	5524	5529	5534	5539	5544	5549	5554	5559	5564	5569	5574	5579	5584	5589	5594	5599	5604	5609	5614	5619	5624	5629	5634	5639	5644	5649	5654	5659	5664	5669	5674	5679	5684	5689	5694	5699	5704	5709	5714	5719	5724	5729	5734	5739	5744	5749	5754	5759	5764	5769	5774	5779	5784	5789	5794	5799	5804	5809	5814	5819	5824	5829	5834	5839	5844	5849	5854	5859	5864	5869	5874	5879	5884	5889	5894	5899	5904	5909	5914	5919	5924	5929	5934	5939	5944	5949	5954	5959	5964	5969	5974	5979	5984	5989	5994	5999	6004	6009	6014	6019	6024	6029	6034	6039	6044	6049	6054	6059	6064	6069	6074	6079	6084	6089	6094	6099	6104	6109	6114	6119	6124	6129	6134	6139	6144	6149	6154	6159	6164	6169	6174	6179	6184	6189	6194	6199	6204	6209	6214	6219	6224	6229	6234	6239	6244	6249	6254	6259	6264	6269	6274	6279	6284	6289	6294	6299	6304	6309	6314	6319	6324	6329	6334	6339	6344	6349	6354	6359	6364	6369	6374	6379	6384	6389	6394	6399	6404	6409	6414	6419	6424	6429	6434	6439	6444	6449	6454	6459	6464	6469	6474	6479	6484	6489	6494	6499	6504	6509	6514	6519	6524	6529	6534	6539	6544	6549	6554	6559	6564	6569	6574	6579	6584	6589	6594	6599	6604	6609	6614	6619	6624	6629	6634	6639	6644	6649	6654	6659	6664	6669	6674	6679	6684	6689	6694	6699	6704	6709	6714	6719	6724	6729	6734	6739	6744	6749	6754	6759	6764	6769	6774	6779	6784	6789	6794	6799	6804	6809	6814	6819	6824	6829	6834	6839	6844	6849	6854	6859	6864	6869	6874	6879	6884	6889	6894	6899	6904	6909	6914	6919	6924	6929	6934	6939	6944	6949	6954	6959	6964	6969	6974	6979	6984	6989	6994	6999	7004	7009	7014	7019	7024	7029	7034	7039	7044	7049	7054	7059	7064	7069	7074	7079	7084	7089	7094	7099	7104	7109	7114	7119	7124	7129	7134	7139	7144	7149	7154	7159	7164	7169	7174	7179	7184	7189	7194	7199	7204	7209	7214	7219	7224	7229	7234	7239	7244	7249	7254	7259	7264	7269	7274	7279	7284	7289	7294	7299	7304	7309	7314	7319	7324	7329	7334	7339	7344	7349	7354	7359	7364	7369	7374	7379	7384	7389	7394	7399	7404	7409	7414	7419	7424	7429	7434	7439	7444	7449	7454	7459	7464	7469	7474	7479	7484	7489	7494	7499	7504	7509	7514	7519	7524	7529	7534	7539	7544	7549	7554	7559	7564	7569	7574	7579	7584	7589	7594	7599	7604	7609	7614	7619	7624	7629	7634	7639	7644	7649	7654	7659	7664	7669	7674	7679	7684	7689	7694	7699	7704	7709	7714	7719	7724	7729	7734	7739	7744	7749	7754	7759	7764	7769	7774	7779	7784	7789	7794	7799	7804	7809	7814	7819	7824	7829	7834	7839	7844	7849	7854	7859	7864	7869	7874	7879	7884	7889	7894	7899	7904	7909	7914	7919	7924	7929	7934	7939	7944	7949	7954	7959	7964	7969	7974	7979	7984	7989	7994	7999	8004	8009	8014	8019	8024	8029	8034	8039	8044	8049	8054	8059	8064	8069	8074	8079	8084	8089	8094	8099	8104	8109	8114	8119	8124	8129	8134	8139	8144	8149	8154	8159	8164	8169	8174	8179	8184	8189	8194	8199	8204	8209	8214	8219	8224	8229	8234	8239	8244	8249	8254	8259	8264	8269	8274	8279	8284	8289	8294	8299	8304	8309	8314	8319	8324	8329	8334	8339	8344	8349	8354	8359	8364	8369	8374	8379	8384	8389	8394	8399	8404	8409	8414	8419	8424	8429	8434	8439	8444	8449	8454	8459	8464	8469	8474	8479	8484	8489	8494	8499	8504	8509	8514	8519	8524	8529	8534	8539	8544	8549	8554	8559	8564	8569	8574	8579	8584	8589	8594	8599	8604	8609	8614	8619	8624	8629	8634	8639	8644	8649	8654	8659	8664	8669	8674	8679	8684	8689	8694	8699	8704	8709	8714	8719	8724	8729	8734	8739	8744	8749	8754	8759	8764	8769	8774	8779	8784	8789	8794	8799	8804	8809	8814	8819	8824	8829	8834	8839	8844	8849	8854	8859	8864	8869	8874	8879	8884	8889	8894	8899	8904	8909	8914	8919	8924	8929	8934	8939	8944	8949	8954	8959	8964	8969	8974	8979	8984	8989	8994	8999	9004	9009	9014	9019	9024	9029	9034	9039	9044	9049	9054	9059	9064	9069	9074	9079	9084	9089	9094	9099	9104	9109	9114	9119	9124	9129	9134	9139	9144	9149	9154	9159	9164	9169	9174	9179	9184	9189	9194	9199	9204	9209	9214	9219	9224	9229	9234	9239	9244	9249	9254	9259	9264	9269	9274	9279	9284	9289	9294	9299	9304	9309	9314	9319	9324	9329	9334	9339	9344	9349	9354	9359	9364	9369	9374	9379	9384	9389	9394	9399	9404	9409	9414	9419	9424	9429	9434	9439	9444	9449	9454	9459	9464	9469	9474	9479	9484	9489	9494	9499	9504	9509	9514	9519	9524	9529	9534	9539	9544	9549	9554	9559	9564	9569	9574	9579	9584	9589	9594	9599	9604	9609	9614	9619	9624	9629	9634	9639	9644	9649	9654	9659	9664	9669	9674	9679	9684	9689	9694	9699	9704	9709	9714	9719	9724	9729	9734	9739	9744	9749	9754	9759	9764	9769	9774	9779	9784	9789	9794	9799	9804	9809	9814	9819	9824	9829	9834	9839	9844	9849	9854	9859	9864	9869	9874	9879	9884	9889	9894	9899	9904	9909	9914	9919	9924	9929	9934	9939	9944	9949	9954	9959	9964	9969	9974	9979	9984	9989	9994	9999	10004	10009	10014	10019	10024	10029	10034	10039	10044	10049	10054	10059	10064	10069	10074	10079	10084	10089	10094	10099	10104	10109	10114	10119	10124	10129	10134	10139	10144	10149	10154	10159	10164	10169	10174	10179	10184	10189	10194	10199	10204	10209	10214	10219	10224	10229	10234	10239	10244	10249	10254	10259	10264	10269	10274	10279	10284	10289	10294	10299	10304	10309	10314	10319	10324	10329	10334	10339	10344	10349	10354	10359	10364	10369	10374	10379	10384	10389	10394	10399	10404	10409	10414	10419	10424	10429	10434	10439	10444	10449	10454	10459	10464	10469	10474	10479	10484	10489	10494	10499	10504	10509	10514	10519	10524	10529	10534	10539	10544	10549	10554	10559	10564	10569	10574	10579	10584	10589	10594	10599	10604	10609	10614	10619	10624	10629	10634	10639	10644	10649	10654	10659	10664	10669	10674	10679	10684	10689	10694	10699	10704	10709	10714	10719	10724	10729	10734	10739	10744	10749	10754	10759	10764	10769	10774	10779	10784	10789	10794	10799	10804	10809	10814	10819	10824	10829	10834	10839	10844	10849	10854	10859	10864	10869	10874	10879	10884	10889	10894	10899	10904	10909	10914	10919	10924	10929	10934	10939	10944	10949	10954	10959	10964	10969	10974	10979	10984	10989	10994	10999	11004	11009	11014	11019	11024	11029	11034	11039	11044	11049	11054	11059	11064	11069	11074	11079	11084	11089	11094	11099	11104	11109	11114	11119	11124	11129	11134	11139	11144	11149	11154	11159	11164	11169	11174	11179	11184	11189	11194	11199	11204	11209	11214	11219	11224	11229	11234	11239	11244	11249	11254	11259	11264	11269	11274	11279	11284	11289	11294	11299	11304	11309	11314	11319	11324	11329	11334	11339	11344	11349	11354	11359	11364	11369	11374	11379	11384	11389	11394	11399	11404	11409	11414	11419	11424	11429	11434	11439	11444	11449	11454	11459	11464	11469	11474	11479	11484	11489	11494	11499	11504	11509	11514	11519	11524	11529	11534	11539	11544	11549	11554	11559	11564	11569	11574	11579	11584	11589	11594	11599	11604	11609	11614	11619	11624	11629	11634	11639	11644	11649	11654	11659	11664	11669	11674	11679	11684	11689	11694	11699	11704	11709	11714	11719	11724	11729	11734	11739	11744	11749	11754	11759	11764	11769	11774	11779	11784	11789	11794	11799	11804	11809	11814	11819	11824	11829	11834	11839	11844	11849	11854	11859	11864	11869	
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--

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
3283	3288	3290	3293	3296	3299	3312	3315	3318	3321	3324	3330	3336
3339	3342	3345	3348	3351	3355	3361	3368	3374	3380	3391	3398	3401
3404	3407	3410	3413	3416	3419	3422	3428	3434	3437	3440	3443	3446
3449	3453	3459	3465	3472	3478	3485	3492	3495	3499	3503	3506	3509
3515	3518	3519	3525	3531	3534	3537	3540	3543	3546	3549	3552	3553
3563	3566	3568	3574	3581	3584	3587	3590	3593	3596	3599	3602	3603
3613	3616	3619	3625	3631	3634	3637	3640	3643	3646	3649	3652	3653
3663	3666	3668	3674	3681	3684	3687	3690	3693	3696	3699	3702	3703
3713	3716	3719	3725	3731	3734	3737	3740	3743	3746	3749	3752	3753
3763	3766	3768	3774	3781	3784	3787	3790	3793	3796	3799	3802	3803
3813	3816	3819	3825	3831	3834	3837	3840	3843	3846	3849	3852	3853
3863	3866	3868	3874	3881	3884	3887	3890	3893	3896	3899	3902	3903
3913	3916	3919	3925	3931	3934	3937	3940	3943	3946	3949	3952	3953
3963	3966	3968	3974	3981	3984	3987	3990	3993	3996	3999	4002	4003
4013	4016	4019	4025	4031	4034	4037	4040	4043	4046	4049	4052	4053
4063	4066	4068	4074	4081	4084	4087	4090	4093	4096	4099	4102	4103
4113	4116	4119	4125	4131	4134	4137	4140	4143	4146	4149	4152	4153
4163	4166	4168	4174	4181	4184	4187	4190	4193	4196	4199	4202	4203
4213	4216	4219	4225	4231	4234	4237	4240	4243	4246	4249	4252	4253
4263	4266	4268	4274	4281	4284	4287	4290	4293	4296	4299	4302	4303
4313	4316	4319	4325	4331	4334	4337	4340	4343	4346	4349	4352	4353
4363	4366	4368	4374	4381	4384	4387	4390	4393	4396	4399	4402	4403
4413	4416	4419	4425	4431	4434	4437	4440	4443	4446	4449	4452	4453
4463	4466	4468	4474	4481	4484	4487	4490	4493	4496	4499	4502	4503
4513	4516	4519	4525	4531	4534	4537	4540	4543	4546	4549	4552	4553
4563	4566	4568	4574	4581	4584	4587	4590	4593	4596	4599	4602	4603
4613	4616	4619	4625	4631	4634	4637	4640	4643	4646	4649	4652	4653
4663	4666	4668	4674	4681	4684	4687	4690	4693	4696	4699	4702	4703
4713	4716	4719	4725	4731	4734	4737	4740	4743	4746	4749	4752	4753
4763	4766	4768	4774	4781	4784	4787	4790	4793	4796	4799	4802	4803
4813	4816	4819	4825	4831	4834	4837	4840	4843	4846	4849	4852	4853
4863	4866	4868	4874	4881	4884	4887	4890	4893	4896	4899	4902	4903
4913	4916	4919	4925	4931	4934	4937	4940	4943	4946	4949	4952	4953
4963	4966	4968	4974	4981	4984	4987	4990	4993	4996	4999	5002	5003
5013	5016	5019	5025	5031	5034	5037	5040	5043	5046	5049	5052	5053
5063	5066	5068	5074	5081	5084	5087	5090	5093	5096	5099	5102	5103
5113	5116	5119	5125	5131	5134	5137	5140	5143	5146	5149	5152	5153
5163	5166	5168	5174	5181	5184	5187	5190	5193	5196	5199	5202	5203
5213	5216	5219	5225	5231	5234	5237	5240	5243	5246	5249	5252	5253
5263	5266	5268	5274	5281	5284	5287	5290	5293	5296	5299	5302	5303
5313	5316	5319	5325	5331	5334	5337	5340	5343	5346	5349	5352	5353
5363	5366	5368	5374	5381	5384	5387	5390	5393	5396	5399	5402	5403
5413	5416	5419	5425	5431	5434	5437	5440	5443	5446	5449	5452	5453
5463	5466	5468	5474	5481	5484	5487	5490	5493	5496	5499	5502	5503
5513	5516	5519	5525	5531	5534	5537	5540	5543	5546	5549	5552	5553
5563	5566	5568	5574	5581	5584	5587	5590	5593	5596	5599	5602	5603
5613	5616	5619	5625	5631	5634	5637	5640	5643	5646	5649	5652	5653
5663	5666	5668	5674	5681	5684	5687	5690	5693	5696	5699	5702	5703
5713	5716	5719	5725	5731	5734	5737	5740	5743	5746	5749	5752	5753
5763	5766	5768	5774	5781	5784	5787	5790	5793	5796	5799	5802	5803
5813	5816	5819	5825	5831	5834	5837	5840	5843	5846	5849	5852	5853
5863	5866	5868	5874	5881	5884	5887	5890	5893	5896	5899	5902	5903
5913	5916	5919	5925	5931	5934	5937	5940	5943	5946	5949	5952	5953
5963	5966	5968	5974	5981	5984	5987	5990	5993	5996	5999	6002	6003
6013	6016	6019	6025	6031	6034	6037	6040	6043	6046	6049	6052	6053
6063	6066	6068	6074	6081	6084	6087	6090	6093	6096	6099	6102	6103
6113	6116	6119	6125	6131	6134	6137	6140	6143	6146	6149	6152	6153
6163	6166	6168	6174	6181	6184	6187	6190	6193	6196	6199	6202	6203
6213	6216	6219	6225	6231	6234	6237	6240	6243	6246	6249	6252	6253
6263	6266	6268	6274	6281	6284	6287	6290	6293	6296	6299	6302	6303
6313	6316	6319	6325	6331	6334	6337	6340	6343	6346	6349	6352	6353
6363	6366	6368	6374	6381	6384	6387	6390	6393	6396	6399	6402	6403
6413	6416	6419	6425	6431	6434	6437	6440	6443	6446	6449	6452	6453
6463	6466	6468	6474	6481	6484	6487	6490	6493	6496	6499	6502	6503
6513	6516	6519	6525	6531	6534	6537	6540	6543	6546	6549	6552	6553
6563	6566	6568	6574	6581	6584	6587	6590	6593	6596	6599	6602	6603
6613	6616	6619	6625	6631	6634	6637	6640	6643	6646	6649	6652	6653
6663	6666	6668	6674	6681	6684	6687	6690	6693	6696	6699	6702	6703
6713	6716	6719	6725	6731	6734	6737	6740	6743	6746	6749	6752	6753
6763	6766	6768	6774	6781	6784	6787	6790	6793	6796	6799	6802	6803
6813	6816	6819	6825	6831	6834	6837	6840	6843	6846	6849	6852	6853
6863	6866	6868	6874	6881	6884	6887	6890	6893	6896	6899	6902	6903
6913	6916	6919	6925	6931	6934	6937	6940	6943	6946	6949	6952	6953
6963	6966	6968	6974	6981	6984	6987	6990	6993	6996	6999	7002	7003
7013	7016	7019	7025	7031	7034	7037	7040	7043	7046	7049	7052	7053
7063	7066	7068	7074	7081	7084	7087	7090	7093	7096	7099	7102	7103
7113	7116	7119	7125	7131	7134	7137	7140	7143	7146	7149	7152	7153
7163	7166	7168	7174	7181	7184	7187	7190	7193	7196	7199	7202	7203
7213	7216	7219	7225	7231	7234	7237	7240	7243	7246	7249	7252	7253
7263	7266	7268	7274	7281	7284	7287	7290	7293	7296	7299	7302	7303
7313	7316	7319	7325	7331	7334	7337	7340	7343	7346	7349	7352	7353
7363	7366	7368	7374	7381	7384	7387	7390	7393	7396	7399	7402	7403
7413	7416	7419	7425	7431	7434	7437	7440	7443	7446	7449	7452	7453
7463	7466	7468	7474	7481	7484	7487	7490	7493	7496	7499	7502	7503
7513	7516	7519	7525	7531	7534	7537	7540	7543	7546	7549	7552	7553
7563	7566	7568	7574	7581	7584	7587	7590	7593	7596	7599	7602	7603
7613	7616	7619	7625	7631	7634	7637	7640	7643	7646	7649	7652	7653
7663	7666	7668	7674	7681	7684	7687	7690	7693	7696	7699	7702	7703
7713	7716	7719	7725	7731	7734	7737	7740	7743	7746	7749	7752	7753
7763	7766	7768	7774	7781	7784	7787	7790	7793	7796	7799	7802	7803
7813	7816	7819	7825	7831	7834	7837	7840	7843	7846	7849	7852	7853
7863	7866	7868	7874	7881	7884	7887	7890	7893	7896	7899	7902	7903
7913	7916	7919	7925	7931	7934	7937	7940	7943	7946	7949	7952	7953
7963	7966	7968	7974	7981	7984	7987	7990	7993	7996	7999	8002	8003
8013	8016	8019	8025	8031	8034	8037	8040	8043	8046	8049	8052	8053
8063	8066	8068	8074	8081	8084	8087	8090	8093	8096	8099	8102	8103
8113	8116	8119	8125	8131	8134</							

7036	7042	7048	7051	7054	7057	7060	7063	7066	7070	7076	7083	7089
7096	7106	7112	7142	7146	7149	7152	7156	7158	7164	7170	7173	7176
7179	7188	7195	7198	7199	7199	7200	7211	7217	7221	7224	7227	7230
7243	7249	7256	7258	7251	7254	7257	7270	7273	7277	7282	7290	7296
7308	7313	7319	7327	7330	7333	7336	7340	7346	7356	7360	7374	7377
7380	7383	7386	7397	7401	7403	7406	7401	7404	7407	7413	7419	7425
7438	7431	7434	7437	7441	7443	7446	7454	7456	7466	7471	7475	7485
7518	7521	7524	7527	7530	7533	7536	7541	7543	7546	7549	7551	7554
7557	7560	7563	7566	7569	7572	7575	7581	7583	7586	7590	7593	7596
7599	7602	7605	7608	7611	7614	7617	7623	7625	7628	7630	7633	7641
7647	7650	7653	7656	7659	7662	7665	7671	7673	7676	7680	7683	7702
7705	7708	7711	7714	7717	7720	7723	7729	7731	7734	7737	7739	7749
7762	7765	7768	7771	7774	7777	7780	7786	7788	7791	7794	7797	7818
7821	7824	7827	7830	7833	7836	7839	7845	7847	7850	7853	7856	7877
7910	7913	7916	7919	7922	7925	7928	7934	7936	7939	7942	7945	7966
7961	7964	7967	7970	7973	7976	7979	7985	7987	7990	7993	7996	8017
8021	8024	8027	8030	8033	8036	8039	8045	8047	8050	8053	8056	8077
8094	8097	8100	8103	8106	8109	8112	8118	8120	8123	8126	8129	8150
8164	8167	8170	8173	8176	8179	8182	8188	8190	8193	8196	8199	8214
8220	8223	8226	8229	8232	8235	8238	8244	8246	8249	8252	8255	8276
8314	8317	8320	8323	8326	8329	8332	8338	8340	8343	8346	8349	8370
8364	8367	8370	8373	8376	8379	8382	8388	8390	8393	8396	8399	8420
8466	8469	8472	8475	8478	8481	8484	8490	8492	8495	8498	8501	8522
8521	8524	8527	8530	8533	8536	8539	8545	8547	8550	8553	8556	8577
8596	8599	8602	8605	8608	8611	8614	8620	8622	8625	8628	8631	8652
8667	8670	8673	8676	8679	8682	8685	8691	8693	8696	8699	8702	8723
8748	8751	8754	8757	8760	8763	8766	8772	8774	8777	8780	8783	8804
8847	8850	8853	8856	8859	8862	8865	8871	8873	8876	8879	8882	8903
8923	8926	8929	8932	8935	8938	8941	8947	8949	8952	8955	8958	8979
9000	9003	9006	9009	9012	9015	9018	9024	9026	9029	9032	9035	9056
9094	9097	9100	9103	9106	9109	9112	9118	9120	9123	9126	9129	9150
9164	9167	9170	9173	9176	9179	9182	9188	9190	9193	9196	9199	9220
9220	9223	9226	9229	9232	9235	9238	9244	9246	9249	9252	9255	9276
9314	9317	9320	9323	9326	9329	9332	9338	9340	9343	9346	9349	9370
9364	9367	9370	9373	9376	9379	9382	9388	9390	9393	9396	9399	9420
9466	9469	9472	9475	9478	9481	9484	9490	9492	9495	9498	9501	9522
9521	9524	9527	9530	9533	9536	9539	9545	9547	9550	9553	9556	9577
9596	9599	9602	9605	9608	9611	9614	9620	9622	9625	9628	9631	9652
9667	9670	9673	9676	9679	9682	9685	9691	9693	9696	9699	9702	9723
9748	9751	9754	9757	9760	9763	9766	9772	9774	9777	9780	9783	9804
8947	8950	8953	8956	8959	8962	8965	8971	8973	8976	8979	8982	8993
9000	9003	9006	9009	9012	9015	9018	9024	9026	9029	9032	9035	9056
9094	9097	9100	9103	9106	9109	9112	9118	9120	9123	9126	9129	9150
9164	9167	9170	9173	9176	9179	9182	9188	9190	9193	9196	9199	9220
9220	9223	9226	9229	9232	9235	9238	9244	9246	9249	9252	9255	9276
9314	9317	9320	9323	9326	9329	9332	9338	9340	9343	9346	9349	9370
9364	9367	9370	9373	9376	9379	9382	9388	9390	9393	9396	9399	9420
9466	9469	9472	9475	9478	9481	9484	9490	9492	9495	9498	9501	9522
9521	9524	9527	9530	9533	9536	9539	9545	9547	9550	9553	9556	9577
9596	9599	9602	9605	9608	9611	9614	9620	9622	9625	9628	9631	9652
9667	9670	9673	9676	9679	9682	9685	9691	9693	9696	9699	9702	9723
9748	9751	9754	9757	9760	9763	9766	9772	9774	9777	9780	9783	9804
9847	9850	9853	9856	9859	9862	9865	9871	9873	9876	9879	9882	9893
9900	9903	9906	9909	9912	9915	9918	9924	9926	9929	9932	9935	9956
9994	9997	10000	10003	10006	10009	10012	10018	10020	10023	10026	10029	10050
10064	10067	10070	10073	10076	10079	10082	10088	10090	10093	10096	10099	10120
10136	10139	10142	10145	10148	10151	10154	10160	10162	10165	10168	10171	10192
10236	10239	10242	10245	10248	10251	10254	10260	10262	10265	10268	10271	10292
10316	10319	10322	10325	10328	10331	10334	10340	10342	10345	10348	10351	10372
10366	10369	10372	10375	10378	10381	10384	10390	10392	10395	10398	10401	10422
10413	10416	10419	10422	10425	10428	10431	10437	10439	10442	10445	10448	10469

CROSS REFERENCE TABLE -- USER SYMBOLS

10467	10471	10477	10484	10490	10496	10511	10516	10519	10522	10556	10558	10561
10514	10570	10573	10576	10579	10582	10584	10587	10591	10594	10597	10600	10603
10505	10509	10512	10515	10518	10521	10524	10527	10530	10533	10536	10539	10542
10545	10548	10551	10554	10557	10560	10563	10566	10569	10572	10575	10578	10581
10584	10587	10590	10593	10596	10599	10602	10605	10608	10611	10614	10617	10620
10623	10626	10629	10632	10635	10638	10641	10644	10647	10650	10653	10656	10659
10662	10665	10668	10671	10674	10677	10680	10683	10686	10689	10692	10695	10698
10701	10704	10707	10710	10713	10716	10719	10722	10725	10728	10731	10734	10737
10740	10743	10746	10749	10752	10755	10758	10761	10764	10767	10770	10773	10776
10779	10782	10785	10788	10791	10794	10797	10800	10803	10806	10809	10812	10815
10818	10821	10824	10827	10830	10833	10836	10839	10842	10845	10848	10851	10854
10857	10860	10863	10866	10869	10872	10875	10878	10881	10884	10887	10890	10893
10896	10899	10902	10905	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	10998	11001	11004	11007	11010
11013	11016	11019	11022	11025	11028	11031	11034	11037	11040	11043	11046	11049
11052	11055	11058	11061	11064	11067	11070	11073	11076	11079	11082	11085	11088
11091	11094	11097	11100	11103	11106	11109	11112	11115	11118	11121	11124	11127
11130	11133	11136	11139	11142	11145	11148	11151	11154	11157	11160	11163	11166
11169	11172	11175	11178	11181	11184	11187	11190	11193	11196	11199	11202	11205
11208	11211	11214	11217	11220	11223	11226	11229	11232	11235	11238	11241	11244
11247	11250	11253	11256	11259	11262	11265	11268	11271	11274	11277	11280	11283
11286	11289	11292	11295	11298	11301	11304	11307	11310	11313	11316	11319	11322
11325	11328	11331	11334	11337	11340	11343	11346	11349	11352	11355	11358	11361
11364	11367	11370	11373	11376	11379	11382	11385	11388	11391	11394	11397	11400
11403	11406	11409	11412	11415	11418	11421	11424	11427	11430	11433	11436	11439
11442	11445	11448	11451	11454	11457	11460	11463	11466	11469	11472	11475	11478
11481	11484	11487	11490	11493	11496	11499	11502	11505	11508	11511	11514	11517
11520	11523	11526	11529	11532	11535	11538	11541	11544	11547	11550	11553	11556
11559	11562	11565	11568	11571	11574	11577	11580	11583	11586	11589	11592	11595
11598	11601	11604	11607	11610	11613	11616	11619	11622	11625	11628	11631	11634
11637	11640	11643	11646	11649	11652	11655	11658	11661	11664	11667	11670	11673
11676	11679	11682	11685	11688	11691	11694	11697	11700	11703	11706	11709	11712
11715	11718	11721	11724	11727	11730	11733	11736	11739	11742	11745	11748	11751
11754	11757	11760	11763	11766	11769	11772	11775	11778	11781	11784	11787	11790
11793	11796	11799	11802	11805	11808	11811	11814	11817	11820	11823	11826	11829
11832	11835	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	11973	11976	11979	11982	11985
11988	11991	11994	11997	12000	12003	12006	12009	12012	12015	12018	12021	12024
12027	12030	12033	12036	12039	12042	12045	12048	12051	12054	12057	12060	12063
12066	12069	12072	12075	12078	12081	12084	12087	12090	12093	12096	12099	12102
12105	12108	12111	12114	12117	12120	12123	12126	12129	12132	12135	12138	12141
12144	12147	12150	12153	12156	12159	12162	12165	12168	12171	12174	12177	12180
12183	12186	12189	12192	12195	12198	12201	12204	12207	12210	12213	12216	12219
12222	12225	12228	12231	12234	12237	12240	12243	12246	12249	12252	12255	12258
12261	12264	12267	12270	12273	12276	12279	12282	12285	12288	12291	12294	12297
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	13607	13610	13613	13616	13619	13622
13625	13628	13631	13634	13637	13640	13643	13646	13649	13652	13655	13658	13661
13664	13667	13670	13673	13676	13679	13682	13685	13688	13691	13694	13697	13700
13703	13706	13709	13712	13715	13718	13721	13724	13727	13730	13733	13736	13739
13742	13745	13748	13751	13754	13757	13760	13763	13766	13769	13772	13775	13778
13781	13784	13787	13790	13793	13796	13799	13802	13805	13808	13811	13814	13817
13820	13823	13826	13829	13832	13835	13838	13841	13844	13847	13850	13853	13856
13859	13862	13865	13868	13871	13874	13877	13880	13883	13886	13889	13892	13895
13898	13901	13904	13907	13910	13913	13916	13919	13922	13925	13928	13931	13934
13937	13940	13943	13946	13949	13952	13955	13					

CROSS REFERENCE TABLE -- USER SYMBOLS

13589	13596	13598	13601	13607	13610	13616	13622	13625	13628	13631	13634	13637
13641	13647	13654	13660	13665	13670	13674	13677	13680	13683	13685	13688	13691
13693	13697	13700	13703	13706	13709	13712	13715	13718	13721	13724	13727	13730
13733	13736	13739	13742	13745	13748	13751	13754	13757	13760	13763	13766	13769
13772	13775	13778	13781	13784	13787	13790	13793	13796	13799	13802	13805	13808
13811	13814	13817	13820	13823	13826	13829	13832	13835	13838	13841	13844	13847
13850	13853	13856	13859	13862	13865	13868	13871	13874	13877	13880	13883	13886
13889	13892	13895	13898	13901	13904	13907	13910	13913	13916	13919	13922	13925
13928	13931	13934	13937	13940	13943	13946	13949	13952	13955	13958	13961	13964
13967	13970	13973	13976	13979	13982	13985	13988	13991	13994	13997	14000	14003
14006	14009	14012	14015	14018	14021	14024	14027	14030	14033	14036	14039	14042
14045	14048	14051	14054	14057	14060	14063	14066	14069	14072	14075	14078	14081
14084	14087	14090	14093	14096	14099	14102	14105	14108	14111	14114	14117	14120
14123	14126	14129	14132	14135	14138	14141	14144	14147	14150	14153	14156	14159
14162	14165	14168	14171	14174	14177	14180	14183	14186	14189	14192	14195	14198
14201	14204	14207	14210	14213	14216	14219	14222	14225	14228	14231	14234	14237
14240	14243	14246	14249	14252	14255	14258	14261	14264	14267	14270	14273	14276
14279	14282	14285	14288	14291	14294	14297	14300	14303	14306	14309	14312	14315
14318	14321	14324	14327	14330	14333	14336	14339	14342	14345	14348	14351	14354
14357	14360	14363	14366	14369	14372	14375	14378	14381	14384	14387	14390	14393
14396	14399	14402	14405	14408	14411	14414	14417	14420	14423	14426	14429	14432
14435	14438	14441	14444	14447	14450	14453	14456	14459	14462	14465	14468	14471
14474	14477	14480	14483	14486	14489	14492	14495	14498	14501	14504	14507	14510
14513	14516	14519	14522	14525	14528	14531	14534	14537	14540	14543	14546	14549
14552	14555	14558	14561	14564	14567	14570	14573	14576	14579	14582	14585	14588
14591	14594	14597	14600	14603	14606	14609	14612	14615	14618	14621	14624	14627
14630	14633	14636	14639	14642	14645	14648	14651	14654	14657	14660	14663	14666
14669	14672	14675	14678	14681	14684	14687	14690	14693	14696	14699	14702	14705
14708	14711	14714	14717	14720	14723	14726	14729	14732	14735	14738	14741	14744
14747	14750	14753	14756	14759	14762	14765	14768	14771	14774	14777	14780	14783
14786	14789	14792	14795	14798	14801	14804	14807	14810	14813	14816	14819	14822
14825	14828	14831	14834	14837	14840	14843	14846	14849	14852	14855	14858	14861
14864	14867	14870	14873	14876	14879	14882	14885	14888	14891	14894	14897	14900
14903	14906	14909	14912	14915	14918	14921	14924	14927	14930	14933	14936	14939
14942	14945	14948	14951	14954	14957	14960	14963	14966	14969	14972	14975	14978
14981	14984	14987	14990	14993	14996	15000	15004	15008	15012	15016	15020	15024
15028	15032	15036	15040	15044	15048	15052	15056	15060	15064	15068	15072	15076
15080	15084	15088	15092	15096	15100	15104	15108	15112	15116	15120	15124	15128
15132	15136	15140	15144	15148	15152	15156	15160	15164	15168	15172	15176	15180
15184	15188	15192	15196	15200	15204	15208	15212	15216	15220	15224	15228	15232
15236	15240	15244	15248	15252	15256	15260	15264	15268	15272	15276	15280	15284
15288	15292	15296	15300	15304	15308	15312	15316	15320	15324	15328	15332	15336
15340	15344	15348	15352	15356	15360	15364	15368	15372	15376	15380	15384	15388
15392	15396	15400	15404	15408	15412	15416	15420	15424	15428	15432	15436	15440
15444	15448	15452	15456	15460	15464	15468	15472	15476	15480	15484	15488	15492
15496	15500	15504	15508	15512	15516	15520	15524	15528	15532	15536	15540	15544
15548	15552	15556	15560	15564	15568	15572	15576	15580	15584	15588	15592	15596
15600	15604	15608	15612	15616	15620	15624	15628	15632	15636	15640	15644	15648
15652	15656	15660	15664	15668	15672	15676	15680	15684	15688	15692	15696	15700
15704	15708	15712	15716	15720	15724	15728	15732	15736	15740	15744	15748	15752
15756	15760	15764	15768	15772	15776	15780	15784	15788	15792	15796	15800	15804
15808	15812	15816	15820	15824	15828	15832	15836	15840	15844	15848	15852	15856
15860	15864	15868	15872	15876	15880	15884	15888	15892	15896	15900	15904	15908
15912	15916	15920	15924	15928	15932	15936	15940	15944	15948	15952	15956	15960
15964	15968	15972	15976	15980	15984	15988	15992	15996	16000	16004	16008	16012
16016	16020	16024	16028	16032	16036	16040	16044	16048	16052	16056	16060	16064
16068	16072	16076	16080	16084	16088	16092	16096	16100	16104	16108	16112	16116
16120	16124	16128	16132	16136	16140	16144	16148	16152	16156	16160	16164	16168
16172	16176	16180	16184	16188	16192	16196	16200	16204	16208	16212	16216	16220
16224	16228	16232	16236	16240	16244	16248	16252	16256	16260	16264	16268	16272
16276	16280	16284	16288	16292	16296	16300	16304	16308	16312	16316	16320	16324
16328	16332	16336	16340	16344	16348	16352	16356	16360	16364	16368	16372	16376
16380	16384	16388	16392	16396	16400	16404	16408	16412	16416	16420	16424	16428
16432	16436	16440	16444	16448	16452	16456	16460	16464	16468	16472	16476	16480
16484	16488	16492	16496	16500	16504	16508	16512	16516	16520	16524	16528	16532
16536	16540	16544	16548	16552	16556	16560	16564	16568	16572	16576	16580	16584
16588	16592	16596	16600	16604	16608	16612	16616	16620	16624	16628	16632	16636
16640	16644	16648	16652	16656	16660	16664	16668	16672	16676	16680	16684	16688
16692	16696	16700	16704	16708	16712	16716	16720	16724	16728	16732	16736	16740
16744	16748	16752	16756	16760	16764	16768	16772	16776	16780	16784	16788	16792
16796	16800	16804	16808	16812	16816	16820	16824	16828	16832	16836	16840	16844
16848	16852	16856	16860	16864	16868	16872	16876	16880	16884	16888	16892	16896
16900	16904	16908	16912	16916	16920	16924	16928	16932	16936	16940	16944	16948
16952	16956	16960	16964	16968	16972	16976	16980	16984	16988	16992	16996	17000
17004	17008	17012	17016	17020	17024	17028	17032	17036	17040	17044	17048	17052
17056	17060	17064	17068	17072	17076	17080	17084	17088	17092	17096	17100	17104
17108	17112	17116	17120	17124	17128	17132	17136	17140	17144	17148	17152	17156
17160	17164	17168	17172	17176	17180	17184	17188	17192	17196	17200	17204	17208
17212	17216	17220	17224	17228	17232	17236	17240	17244	17248	17252	17256	17260
17264	17268	17272	17276	17280	17284	17288	17292	17296	17300	17304	17308	17312
17316	17320	17324	17328	17332	17336	17340	17344	17348	17352	17356	17360	17364
17368	17372	17376	17380	17384	17388	17392	17396	17400	17404	17408	17412	17416
17420	17424	17428	17432	17436	17440	17444	17448	17452	17456	17460	17464	17468
17472	17476	17480	17484	17488	17492	17496	17500	17504	17508	17512	17516	17520
17524	17528	17532	17536	17540	17544	17548	17552	17556	17560	17564	17568	17572
17576	17580	17584	17588	17592	17596	17600	17604	17608	17612	17616	17620	17624
17628	17632	17636	17640	17644	17648	17652	17656	17660	17664	17668	17672	17676
17680	17684	17688	17692	17696	17700	17704	17708	17712	17716	17720	17724	17728
17732	17736	17740	17744	17748	17752	17756	17760	17764	17768	17772	17776	17780
17784	17788	17792	17796	17800	17804	17808	17812	17816	17820	17824	1782	

CROSS REFERENCE TABLE -- USER SYMBOLS

3538	3539	3541	3551	3553	3554	3556	3557	3559	3560	3562	3563	3565
3540	3541	3543	3552	3576	3578	3583	3585	3589	3591	3595	3597	3609
3541	3542	3544	3553	3577	3579	3621	3631	3633	3634	3636	3637	3639
3542	3543	3545	3554	3578	3580	3650	3652	3655	3658	3663	3665	3669
3543	3544	3546	3555	3579	3581	3721	3722	3725	3728	3728	3730	3732
3544	3545	3547	3556	3580	3582	3750	3753	3754	3756	3757	3759	3760
3545	3546	3548	3557	3581	3583	3773	3780	3782	3785	3788	3792	3794
3546	3547	3549	3558	3582	3584	3816	3818	3819	3820	3821	3823	3824
3547	3548	3550	3559	3583	3585	3916	3918	3919	3920	3921	3923	3924
3548	3549	3551	3560	3584	3586	3947	3948	3949	3950	3951	3953	3954
3549	3550	3552	3561	3585	3587	3970	3972	3977	3979	3980	3982	3983
3550	3551	3553	3562	3586	3588	4012	4013	4015	4017	4018	4020	4021
3551	3552	3554	3563	3587	3589	4040	4042	4044	4046	4047	4049	4050
3552	3553	3555	3564	3588	3590	4082	4113	4115	4116	4118	4120	4122
3553	3554	3556	3565	3589	3591	4142	4144	4145	4147	4148	4150	4151
3554	3555	3557	3566	3590	3592	4163	4167	4169	4174	4176	4180	4182
3555	3556	3558	3567	3591	3593	4207	4209	4210	4212	4222	4224	4225
3556	3557	3559	3568	3592	3594	4236	4237	4239	4241	4243	4247	4249
3557	3558	3560	3569	3593	3595	4277	4279	4310	4312	4313	4315	4317
3558	3559	3561	3570	3594	3596	4303	4305	4311	4314	4314	4315	4317
3559	3560	3562	3571	3595	3597	4333	4335	4341	4343	4344	4345	4347
3560	3561	3563	3572	3596	3598	4366	4368	4364	4366	4371	4373	4377
3561	3562	3564	3573	3597	3599	4406	4408	4407	4407	4409	4419	4421
3562	3563	3565	3574	3598	3600	4434	4436	4436	4436	4438	4440	4444
3563	3564	3566	3575	3599	3601	4466	4468	4476	4476	4479	4480	4482
3564	3565	3567	3576	3600	3602	4507	4509	4507	4508	4509	4510	4512
3565	3566	3568	3577	3601	3603	4536	4538	4537	4538	4539	4541	4542
3566	3567	3569	3578	3602	3604	4567	4569	4567	4568	4562	4568	4570
3567	3568	3570	3579	3603	3605	4593	4595	4591	4592	4593	4596	4606
3568	3569	3571	3580	3604	3606	4623	4625	4623	4624	4623	4626	4637
3569	3570	3572	3581	3605	3607	4650	4652	4650	4651	4653	4656	4667
3570	3571	3573	3582	3606	3608	4677	4679	4677	4678	4673	4674	4677
3571	3572	3574	3583	3607	3609	4702	4704	4702	4703	4704	4706	4707
3572	3573	3575	3584	3608	3610	4723	4725	4723	4724	4725	4726	4728
3573	3574	3576	3585	3609	3611	4750	4752	4750	4751	4752	4754	4755
3574	3575	3577	3586	3610	3612	4777	4779	4777	4778	4779	4781	4782
3575	3576	3578	3587	3611	3613	4802	4804	4802	4803	4804	4806	4807
3576	3577	3579	3588	3612	3614	4827	4829	4827	4828	4829	4831	4832
3577	3578	3580	3589	3613	3615	4854	4856	4854	4855	4856	4858	4859
3578	3579	3581	3590	3614	3616	4881	4883	4881	4882	4883	4885	4886
3579	3580	3582	3591	3615	3617	4912	4914	4912	4913	4914	4916	4917
3580	3581	3583	3592	3616	3618	4937	4939	4937	4938	4939	4941	4942
3581	3582	3584	3593	3617	3619	4964	4966	4964	4965	4966	4968	4969
3582	3583	3585	3594	3618	3620	4991	4993	4991	4992	4993	4995	4996
3583	3584	3586	3595	3619	3621	5012	5014	5012	5013	5014	5016	5017
3584	3585	3587	3596	3620	3622	5037	5039	5037	5038	5039	5041	5042
3585	3586	3588	3597	3621	3623	5064	5066	5064	5065	5066	5068	5069
3586	3587	3589	3598	3622	3624	5091	5093	5091	5092	5093	5095	5096
3587	3588	3590	3599	3623	3625	5112	5114	5112	5113	5114	5116	5117
3588	3589	3591	3600	3624	3626	5137	5139	5137	5138	5139	5141	5142
3589	3590	3592	3601	3625	3627	5164	5166	5164	5165	5166	5168	5169
3590	3591	3593	3602	3626	3628	5191	5193	5191	5192	5193	5195	5196
3591	3592	3594	3603	3627	3629	5212	5214	5212	5213	5214	5216	5217
3592	3593	3595	3604	3628	3630	5237	5239	5237	5238	5239	5241	5242
3593	3594	3596	3605	3629	3631	5264	5266	5264	5265	5266	5268	5269
3594	3595	3597	3606	3630	3632	5291	5293	5291	5292	5293	5295	5296
3595	3596	3598	3607	3631	3633	5312	5314	5312	5313	5314	5316	5317
3596	3597	3599	3608	3632	3634	5337	5339	5337	5338	5339	5341	5342
3597	3598	3600	3609	3633	3635	5364	5366	5364	5365	5366	5368	5369
3598	3599	3601	3610	3634	3636	5391	5393	5391	5392	5393	5395	5396
3599	3600	3602	3611	3635	3637	5412	5414	5412	5413	5414	5416	5417
3600	3601	3603	3612	3636	3638	5437	5439	5437	5438	5439	5441	5442
3601	3602	3604	3613	3637	3639	5464	5466	5464	5465	5466	5468	5469
3602	3603	3605	3614	3638	3640	5491	5493	5491	5492	5493	5495	5496
3603	3604	3606	3615	3639	3641	5512	5514	5512	5513	5514	5516	5517
3604	3605	3607	3616	3640	3642	5537	5539	5537	5538	5539	5541	5542
3605	3606	3608	3617	3641	3643	5564	5566	5564	5565	5566	5568	5569
3606	3607	3609	3618	3642	3644	5591	5593	5591	5592	5593	5595	5596
3607	3608	3610	3619	3643	3645	5612	5614	5612	5613	5614	5616	5617
3608	3609	3611	3620	3644	3646	5637	5639	5637	5638	5639	5641	5642
3609	3610	3612	3621	3645	3647	5664	5666	5664	5665	5666	5668	5669
3610	3611	3613	3622	3646	3648	5691	5693	5691	5692	5693	5695	5696
3611	3612	3614	3623	3647	3649	5712	5714	5712	5713	5714	5716	5717
3612	3613	3615	3624	3648	3650							
3613	3614	3616	3625	3649	3651							
3614	3615	3617	3626	3650	3652							
3615	3616	3618	3627	3651	3653							
3616	3617	3619	3628	3652	3654							
3617	3618	3620	3629	3653	3655							
3618	3619	3621	3630	3654	3656							
3619	3620	3622	3631	3655	3657							
3620	3621	3623	3632	3656	3658							
3621	3622	3624	3633	3657	3659							
3622	3623	3625	3634	3658	3660							
3623	3624	3626	3635	3659	3661							
3624	3625	3627	3636	3660	3662							
3625	3626	3628	3637	3661	3663							
3626	3627	3629	3638	3662	3664							
3627	3628	3630	3639	3663	3665							
3628	3629	3631	3640	3664	3666							
3629	3630	3632	3641	3665	3667							
3630	3631	3633	3642	3666	3668							
3631	3632	3634	3643	3667	3669							
3632	3633	3635	3644	3668	3670							
3633	3634	3636	3645	3669	3671							
3634	3635	3637	3646	3670	3672							
3635	3636	3638	3647	3671	3673							
3636	3637	3639	3648	3672	3674							
3637	3638	3640	3649	3673	3675							
3638	3639	3641	3650	3674	3676							
3639	3640	3642	3651	3675	3677							
3640	3641	3643	3652	3676	3678							
3641	3642	3644	3653	3677	3679							
3642	3643	3645	3654	3678	3680							
3643	3644	3646	3655	3679	3681							
3644	3645	3647	3656	3680	3682							
3645	3646	3648	3657	3681	3683							
3646	3647	3649	3658	3682	3684							
3647	3648	3650	3659	3683	3685							
3648	3649	3651	3660	3684	3686							
3649	3650	3652	3661	3685	3687							
3650	3651	3653	3662									

CROSS REFERENCE TABLE -- USER SYMBOLS

5708	5709	5711	5721	5723	5724	5726	5727	5729	5730	5732	5733	5735
5736	5738	5739	5741	5743	5745	5748	5751	5756	5758	5762	5764	5768
5770	5782	5784	5785	5787	5788	5790	5791	5793	5794	5796	5806	5808
5809	5811	5812	5814	5815	5817	5818	5820	5821	5823	5824	5825	5828
5830	5834	5836	5841	5843	5847	5849	5850	5855	5856	5857	5859	5859
5900	5902	5903	5906	5907	5909	5910	5912	5913	5915	5916	5918	5928
5930	5931	5932	5934	5935	5937	5939	5940	5942	5943	5945	5946	5948
5950	5951	5952	5953	5955	5956	5957	5971	5972	5977	5978	5991	5992
6022	6024	6025	6027	6030	6031	6033	6033	6035	6037	6038	6043	6048
6050	6054	6055	6056	6057	6071	6073	6104	6105	6107	6109	6111	6113
6114	6116	6117	6119	6120	6122	6123	6125	6126	6127	6128	6140	6141
6143	6144	6146	6147	6149	6150	6151	6152	6155	6157	6159	6163	6165
6170	6172	6176	6178	6180	6184	6185	6188	6192	6201	6202	6204	6205
6207	6208	6210	6220	6222	6223	6224	6225	6228	6229	6231	6232	6234
6235	6237	6238	6240	6242	6244	6248	6250	6251	6257	6258	6263	6267
6269	6270	6271	6311	6312	6313	6314	6316	6320	6321	6323	6324	6325
6327	6328	6330	6331	6332	6333	6334	6335	6337	6350	6351	6353	6354
6355	6356	6358	6359	6360	6361	6362	6363	6364	6372	6379	6383	6385
6386	6387	6388	6389	6390	6391	6392	6393	6394	6411	6412	6417	6427
6428	6429	6430	6431	6432	6433	6434	6435	6436	6437	6441	6445	6447
6449	6450	6451	6452	6453	6454	6455	6456	6457	6470	6474	6487	6518
6520	6521	6522	6523	6524	6525	6526	6527	6528	6531	6533	6537	6539
6543	6544	6545	6546	6547	6548	6549	6550	6551	6552	6553	6556	6567
6569	6571	6573	6577	6579	6581	6582	6583	6584	6585	6586	6588	6612
6613	6615	6616	6618	6619	6648	6649	6651	6652	6654	6658	6660	6662
6642	6643	6645	6646	6647	6648	6649	6650	6651	6652	6653	6662	6664
6669	6671	6675	6677	6681	6682	6683	6684	6685	6686	6687	6688	6689
6734	6735	6737	6738	6740	6741	6743	6744	6745	6752	6758	6759	6761
6762	6764	6765	6767	6768	6770	6771	6773	6774	6776	6778	6780	6784
6786	6791	6793	6797	6799	6803	6805	6817	6819	6820	6822	6823	6825
6826	6828	6829	6831	6841	6843	6844	6845	6847	6849	6850	6852	6853
6855	6856	6858	6859	6861	6863	6865	6867	6871	6876	6878	6882	6884
6888	6890	6899	6901	6922	6934	6935	6937	6939	6941	6942	6944	6945
6947	6948	6950	6951	6953	6953	6955	6956	6958	6959	6971	6972	6974
6975	6977	6978	6980	6981	6983	6985	6987	6991	6993	6998	7000	7004
7006	7010	7012	7024	7026	7027	7029	7030	7032	7033	7035	7036	7038
7048	7050	7051	7053	7054	7056	7057	7059	7060	7062	7063	7065	7066
7068	7070	7072	7076	7078	7083	7085	7089	7091	7095	7097	7108	7108
7139	7141	7142	7144	7146	7148	7149	7151	7152	7154	7155	7157	7158
7160	7170	7172	7173	7175	7176	7178	7179	7181	7182	7184	7185	7187
7188	7190	7192	7194	7198	7200	7205	7207	7211	7213	7217	7219	7231
7233	7234	7236	7237	7239	7240	7242	7243	7245	7255	7257	7258	7260
7261	7263	7264	7266	7267	7269	7270	7272	7273	7275	7277	7279	7280
7285	7290	7292	7296	7298	7302	7304	7313	7315	7344	7346	7347	7349
7350	7352	7353	7355	7356	7358	7359	7361	7362	7364	7365	7367	7368
7370	7374	7376	7377	7379	7380	7382	7383	7385	7386	7388	7389	7391
7392	7394	7395	7397	7398	7400	7401	7403	7404	7406	7407	7409	7413
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	7485
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	7557
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

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	7829	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	7999	8017
8018	8020	8021	8023	8024	8026	8027	8029	8030	8032	8033	8035	8036
8038	8039	8041	8042	8044	8045	8047	8048	8050	8060	8062	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	8174	8184	8186	8187	8188	8190	8191
8193	8194	8196	8197	8199	8201	8203	8207	8209	8214	8216	8220	8222
8226	8228	8241	8243	8244	8246	8247	8259	8264	8266	8268	8298	8299
8301	8302	8304	8306	8307	8308	8310	8311	8313	8314	8316	8317	8319
8321	8323	8324	8326	8327	8328	8329	8341	8342	8344	8345	8348	8349
8351	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	8526	8528	8529	8534	8535	8537	8538	8539
8561	8563	8564	8566	8567	8569	8579	8581	8582	8584	8585	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	8788	8788	8803	8805	8837	8838	8840	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	9243	9245	9278	9280	9281	9283	9284	9286
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	9336	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

K10

CROSS REFERENCE TABLE -- USER SYMBOLS

1455	1454	1466	1469	1470	1470	1470	1470	1470	1470	1470	1471	1471	1471
1471	1471	1471	1471	1472	1472	1472	1472	1472	1472	1472	1473	1473	1473
1474	1475	1475	1475	1476	1476	1476	1476	1476	1476	1476	1477	1477	1477
1482	1482	1482	1482	1483	1483	1483	1483	1483	1483	1483	1484	1484	1484
1482	1482	1482	1482	1483	1483	1483	1483	1483	1483	1483	1484	1484	1484
1490	1490	1490	1490	1490	1491	1491	1491	1491	1491	1491	1492	1492	1492
1493	1493	1491	1491	1491	1491	1491	1491	1491	1491	1491	1492	1492	1492
1493	1493	1491	1491	1491	1491	1491	1491	1491	1491	1491	1492	1492	1492
1497	1497	1497	1497	1497	1497	1497	1497	1497	1497	1497	1498	1498	1498
1501	1501	1501	1501	1501	1501	1501	1501	1501	1501	1501	1502	1502	1502
1503	1503	1501	1501	1501	1501	1501	1501	1501	1501	1501	1502	1502	1502
1016	203	205	208	221	222	224	226	228	232	236	246	294	380
381	382	383	392	392	480	486	487	489	491	534	536	537	538
539	541	542	543	544	546	547	548	549	551	552	553	554	554
556	557	558	559	561	562	563	564	566	567	568	569	571	571
572	573	574	576	577	578	579	581	582	583	584	586	587	587
588	589	591	592	593	594	596	597	598	599	601	602	603	603
604	606	607	608	609	611	612	613	614	615	616	617	618	618
1010	1081	1082	1174	1226	1236	1236	1328	1329	1335	1389	1678	1707	1750
1769	1785	1787	1841	1844	1855	1864	1864	2005	2088	2092	2138	2169	2202
2205	2281	2478	2675	2872	3070	3287	3520	3717	3914	4111	4308	4505	4505
4702	4899	5097	5274	5481	5688	5895	6102	6309	6516	6723	6930	7137	7137
7343	7517	7760	7977	8137	8294	8455	8668	8834	8950	9277	9594	9911	9911
10228	10545	10862	11179	11496	11813	12130	12447	12764	13081	13398	13713	13856	13856
14480	14698	14808	15310	15312	8009	8097	8238	8398	8533	8617	8696	8783	8865
1	5162	5232	8009	8097	8238	8398	8533	8617	8696	8783	8865	8916	8916
12905	12973	13792	13968	14770	14773	9483	9548	9729	9800	9865	10046	10117	10182
1	9095	9166	9231	9412	9483	9548	9729	9800	9865	10046	10117	10182	10182
10363	10434	10499	10680	10686	10751	10754	10816	10997	11068	11133	11314	11385	11385
11450	11631	11702	11767	11948	12019	12084	12265	12336	12401	12582	12653	12718	12718
12899	12970	13035	13216	13287	13352	13533	13604	13669	14863	14933	14933	14933	14933
1	11637	11705	3181	3309	3407	7861	9735	9803	14535	14836	14906	14970	14991
1429	1569	3092	7617	7644	7827	7988	8871	8922	14571	14854	15037	14970	14991
1	3092	7410	7617	7644	7827	7988	8871	8922	14571	14854	15037	14970	14991
1	14577	14592	14924	14924	14559	14741	2836	2953	3033	3160	3249	3386	3484
1	13732	13760	13786	14559	14741	2836	2953	3033	3160	3249	3386	3484	3484
1	2362	2442	2559	2639	2756	2836	2953	3033	3160	3249	3386	3484	3484
3601	3681	3798	3878	3958	4075	4192	4272	4389	4469	4586	4666	4783	4783
4863	4980	5060	5168	5238	5360	5445	5567	5652	5774	5859	5981	6066	6066
6188	6273	6395	6480	6602	6687	6809	6894	7016	7101	7223	7308	8232	8232
8389	13798	14547	14735	14761	14848	14857	14918	14927	14982	14997	15031	14976	14976
1	7833	8250	8410	8545	8629	8705	8795	13983	14015	14392	14776	14776	14776
1	2334	2340	2347	2353	2365	2414	2420	2427	2433	2450	2453	2531	2531
2537	2544	2550	2562	2611	2617	2624	2630	2647	2650	2728	2734	2741	2741
2747	2759	2808	2814	2821	2827	2844	2847	2855	2861	2931	2944	2956	2956
3005	3011	3018	3024	3041	3044	3132	3138	3145	3151	3163	3221	3227	3227
3234	3240	3257	3260	3268	3264	3371	3377	3389	3456	3462	3469	3475	3475
3492	3495	3573	3579	3586	3592	3604	3653	3659	3666	3672	3689	3692	3692
3770	3776	3783	3789	3801	3850	3855	3863	3869	3886	3889	3967	3973	3973
3980	3986	3998	4047	4053	4060	4066	4083	4086	4164	4170	4177	4183	4183
4195	4244	4250	4257	4263	4280	4283	4361	4367	4374	4380	4392	4441	4441
4447	4454	4460	4477	4480	4488	4494	4571	4577	4589	4638	4644	4651	4651
4657	4674	4677	4755	4761	4768	4774	4786	4835	4841	4848	4854	4871	4871
4874	4952	4958	4965	4971	4983	5032	5038	5045	5051	5068	5071	5140	5140

\$40CAT= ***** U
= 037470

.ADC = 000100

.A00 = 000000

.A00MC= 000020

.A00M = 006050

.A00 = 000260

.B80 = 002000

.B81 = 002400

.B84 = 003000

.B87 = 003400

.9C = 001000

.BEGIN = 003464

.BR = 000400

CROSS REFERENCE TABLE -- USER SYMBOLS

6282	6285	6294	6298	6374	6381	6387	6396	6399	6425	6453	6459	6466
6472	6481	6486	6488	6547	6575	6581	6588	6594	6603	6606	6632	6660
6666	6673	6677	6679	6696	6699	6754	6782	6788	6795	6801	6810	6813
6839	6847	6852	6854	6866	6869	6903	6906	6961	6999	6995	7002	7008
7017	7020	7026	7027	7080	7087	7093	7102	7110	7113	7168	7196	7202
7209	7215	7220	7222	7233	7237	7241	7242	7294	7300	7309	7317	7320
7423	7445	7452	7456	7479	7482	7489	7492	7618	7645	7660	7682	7685
7700	7722	7725	7732	7738	7739	7821	7827	7850	7853	7869	7872	7956
7959	7999	8001	8013	8028	8030	8083	8089	8101	8108	8111	8180	8205
8211	8218	8221	8233	8251	8258	8268	8269	8337	8362	8368	8375	8381
8390	8393	8411	8421	8432	8431	8487	8487	8518	8525	8528	8546	8549
8577	8579	8582	8592	8609	8630	8633	8641	8644	8706	8709	8737	8759
8765	8772	8778	8782	8792	8807	8810	8872	8886	8889	8896	8899	8923
8930	8933	8939	8941	8942	8945	8953	8957	9155	9207	9213	9226	9237
9240	9247	9251	9252	9253	9254	9257	9257	9182	9499	9524	9537	9543
9554	9557	9558	9559	9562	9563	9567	9567	9472	9789	9816	9847	9854
9860	9871	9874	9881	9884	9884	9770	9776	9783	9816	9816	9841	9854
10171	10177	10188	10191	10191	10201	10201	10203	10203	10106	10106	10158	10164
10481	10488	10491	10505	10505	10505	10505	10505	10404	10410	10417	10450	10475
10742	10749	10805	10811	10811	10811	10811	10811	10518	10721	10727	10740	10767
11094	11109	11115	11123	11123	11123	11123	11123	10826	11013	11038	11051	11057
11374	11401	11426	11432	11432	11432	11432	11432	11149	11152	11330	11361	11368
11685	11691	11718	11743	11743	11743	11743	11743	11459	11462	11469	11672	11678
11995	12002	12008	12036	12036	12036	12036	12036	11773	11776	11783	11964	11989
12306	12312	12319	12324	12324	12324	12324	12324	12079	12090	12093	12103	12281
12598	12623	12634	12636	12636	12636	12636	12636	12390	12396	12407	12410	12420
12737	12915	12940	12946	12946	12946	12946	12946	12700	12707	12713	12727	12734
13051	13054	13057	13057	13057	13057	13057	13057	13011	13017	13024	13041	13044
13361	13368	13371	13371	13371	13371	13371	13371	13203	13228	13234	13341	13358
13675	13678	13685	13688	13688	13688	13688	13688	13593	13620	13645	13651	13664
13802	13816	13819	13826	13826	13826	13826	13826	13761	13754	13778	13781	13799
14041	14057	14057	14057	14057	14057	14057	14057	13994	14000	14007	14016	14025
14179	14189	14199	14215	14234	14247	14247	14247	14117	14129	14139	14149	14169
14355	14384	14387	14393	14396	14402	14405	14412	14273	14286	14299	14312	14338
14572	14578	14587	14593	14602	14617	14620	14636	14260	14415	14430	14452	14560
14742	14762	14765	14777	14780	14788	14820	14836	14649	14658	14661	14671	14736
14998	15032	15038	15045	15051	15051	15051	15051	14855	14858	14919	14928	14983
18	18	2309	2389	2506	2586	2703	2900	2783	2900	3107	3196	3431
3548	3628	3745	3825	3942	4022	4139	4219	4139	4336	4416	4533	4730
4810	4927	5007	5124	5194	5304	5389	5511	5389	5536	5718	5803	6010
6132	6217	6339	6424	6546	6631	6753	6838	6753	6960	7045	7167	7422
7459	7559	7699	7855	8000	8057	8088	8179	8088	8336	8486	8576	9110
9181	9236	9427	9498	9553	9744	9815	9870	9815	10061	10132	10187	10449
10504	10695	10766	10821	11012	11083	11138	11329	10661	11400	11455	11646	11772
11963	12034	12089	12280	12351	12406	12597	12668	12597	12723	12914	12985	13201
13302	13357	13548	13619	13674	14006	14364	14401	14364	14429	14616	14635	
14628	1568											
14618	1567											
1564	1741	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	2664	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

.CHVRT 006160
.CONVR 006154
.CO = 000400
.DATAC 007420
.DBR = 000400

CROSS REFERENCE TABLE -- USER SYMBOLS

3072	3079	3092	3104	3129	3136	3142	3148	3154	3168	3181	3193	3218
3071	3070	3091	3103	3128	3135	3141	3147	3153	3167	3180	3192	3217
3070	3069	3090	3102	3127	3134	3140	3146	3152	3166	3179	3191	3216
3069	3068	3089	3101	3126	3133	3139	3145	3151	3165	3178	3190	3215
3068	3067	3088	3100	3125	3132	3138	3144	3150	3164	3177	3189	3214
3067	3066	3087	3099	3124	3131	3137	3143	3149	3163	3176	3188	3213
3066	3065	3086	3098	3123	3130	3136	3142	3148	3162	3175	3187	3212
3065	3064	3085	3097	3122	3129	3135	3141	3147	3161	3174	3186	3211
3064	3063	3084	3096	3121	3128	3134	3140	3146	3160	3173	3185	3210
3063	3062	3083	3095	3120	3127	3133	3139	3145	3159	3172	3184	3209
3062	3061	3082	3094	3119	3126	3132	3138	3144	3158	3171	3183	3208
3061	3060	3081	3093	3118	3125	3131	3137	3143	3157	3170	3182	3207
3060	3059	3080	3092	3117	3124	3130	3136	3142	3156	3169	3181	3206
3059	3058	3079	3091	3116	3123	3129	3135	3141	3155	3168	3180	3205
3058	3057	3078	3090	3115	3122	3128	3134	3140	3154	3167	3179	3204
3057	3056	3077	3089	3114	3121	3127	3133	3139	3153	3166	3178	3203
3056	3055	3076	3088	3113	3120	3126	3132	3138	3152	3165	3177	3202
3055	3054	3075	3087	3112	3119	3125	3131	3137	3151	3164	3176	3201
3054	3053	3074	3086	3111	3118	3124	3130	3136	3150	3163	3175	3200
3053	3052	3073	3085	3110	3117	3123	3129	3135	3149	3162	3174	3199
3052	3051	3072	3084	3109	3116	3122	3128	3134	3148	3161	3173	3198
3051	3050	3071	3083	3108	3115	3121	3127	3133	3147	3160	3172	3197
3050	3049	3070	3082	3107	3114	3120	3126	3132	3146	3159	3171	3196
3049	3048	3069	3081	3106	3113	3119	3125	3131	3145	3158	3170	3195
3048	3047	3068	3080	3105	3112	3118	3124	3130	3144	3157	3169	3194
3047	3046	3067	3079	3104	3111	3117	3123	3129	3143	3156	3168	3193
3046	3045	3066	3078	3103	3110	3116	3122	3128	3142	3155	3167	3192
3045	3044	3065	3077	3102	3109	3115	3121	3127	3141	3154	3166	3191
3044	3043	3064	3076	3101	3108	3114	3120	3126	3140	3153	3165	3190
3043	3042	3063	3075	3100	3107	3113	3119	3125	3139	3152	3164	3189
3042	3041	3062	3074	3099	3106	3112	3118	3124	3138	3151	3163	3188
3041	3040	3061	3073	3098	3105	3111	3117	3123	3137	3150	3162	3187
3040	3039	3060	3072	3097	3104	3110	3116	3122	3136	3149	3161	3186
3039	3038	3059	3071	3096	3103	3109	3115	3121	3135	3148	3160	3185
3038	3037	3058	3070	3095	3102	3108	3114	3120	3134	3147	3159	3184
3037	3036	3057	3069	3094	3101	3107	3113	3119	3133	3146	3158	3183
3036	3035	3056	3068	3093	3100	3106	3112	3118	3132	3145	3157	3182
3035	3034	3055	3067	3092	3099	3105	3111	3117	3131	3144	3156	3181
3034	3033	3054	3066	3091	3098	3104	3110	3116	3130	3143	3155	3180
3033	3032	3053	3065	3090	3097	3103	3109	3115	3129	3142	3154	3179
3032	3031	3052	3064	3089	3096	3102	3108	3114	3128	3141	3153	3178
3031	3030	3051	3063	3088	3095	3101	3107	3113	3127	3140	3152	3177
3030	3029	3050	3062	3087	3094	3100	3106	3112	3126	3139	3151	3176
3029	3028	3049	3061	3086	3093	3099	3105	3111	3125	3138	3150	3175
3028	3027	3048	3060	3085	3092	3098	3104	3110	3124	3137	3149	3174
3027	3026	3047	3059	3084	3091	3097	3103	3109	3123	3136	3148	3173
3026	3025	3046	3058	3083	3090	3096	3102	3108	3122	3135	3147	3172
3025	3024	3045	3057	3082	3089	3095	3101	3107	3121	3134	3146	3171
3024	3023	3044	3056	3081	3088	3094	3100	3106	3120	3133	3145	3170
3023	3022	3043	3055	3080	3087	3093	3099	3105	3119	3132	3144	3169
3022	3021	3042	3054	3079	3086	3092	3098	3104	3118	3131	3143	3168
3021	3020	3041	3053	3078	3085	3091	3097	3103	3117	3130	3142	3167
3020	3019	3040	3052	3077	3084	3090	3096	3102	3116	3129	3141	3166
3019	3018	3039	3051	3076	3083	3089	3095	3101	3115	3128	3140	3165
3018	3017	3038	3050	3075	3082	3088	3094	3100	3114	3127	3139	3164
3017	3016	3037	3049	3074	3081	3087	3093	3099	3113	3126	3138	3163
3016	3015	3036	3048	3073	3080	3086	3092	3098	3112	3125	3137	3162
3015	3014	3035	3047	3072	3079	3085	3091	3097	3111	3124	3136	3161
3014	3013	3034	3046	3071	3078	3084	3090	3096	3110	3123	3135	3160
3013	3012	3033	3045	3070	3077	3083	3089	3095	3109	3122	3134	3159
3012	3011	3032	3044	3069	3076	3082	3088	3094	3108	3121	3133	3158
3011	3010	3031	3043	3068	3075	3081	3087	3093	3107	3120	3132	3157
3010	3009	3030	3042	3067	3074	3080	3086	3092	3106	3119	3131	3156
3009	3008	3029	3041	3066	3073	3079	3085	3091	3105	3118	3130	3155
3008	3007	3028	3040	3065	3072	3078	3084	3090	3104	3117	3129	3154
3007	3006	3027	3039	3064	3071	3077	3083	3089	3103	3116	3128	3153
3006	3005	3026	3038	3063	3070	3076	3082	3088	3102	3115	3127	3152
3005	3004	3025	3037	3062	3069	3075	3081	3087	3101	3114	3126	3151
3004	3003	3024	3036	3061	3068	3074	3080	3086	3100	3113	3125	3150
3003	3002	3023	3035	3060	3067	3073	3079	3085	3099	3112	3124	3149
3002	3001	3022	3034	3059	3066	3072	3078	3084	3098	3111	3123	3148
3001	3000	3021	3033	3058	3065	3071	3077	3083	3097	3110	3122	3147
3000	2999	3020	3032	3057	3064	3070	3076	3082	3096	3109	3121	3146
2999	2998	3019	3031	3056	3063	3069	3075	3081	3095	3108	3120	3145
2998	2997	3018	3030	3055	3062	3068	3074	3080	3094	3107	3119	3144
2997	2996	3017	3029	3054	3061	3067	3073	3079	3093	3106	3118	3143
2996	2995	3016	3028	3053	3060	3066	3072	3078	3092	3105	3117	3142
2995	2994	3015	3027	3052	3059	3065	3071	3077	3091	3104	3116	3141
2994	2993	3014	3026	3051	3058	3064	3070	3076	3090	3103	3115	3140
2993	2992	3013	3025	3050	3057	3063	3069	3075	3089	3102	3114	3139
2992	2991	3012	3024	3049	3056	3062	3068	3074	3088	3101	3113	3138
2991	2990	3011	3023	3048	3055	3061	3067	3073	3087	3100	3112	3137
2990	2989	3010	3022	3047	3054	3060	3066	3072	3086	3099	3111	3136
2989	2988	3009	3021	3046	3053	3059	3065	3071	3085	3098	3110	3135
2988	2987	3008	3020	3045	3052	3058	3064	3070	3084	3097	3109	3134
2987	2986	3007	3019	3044	3051	3057	3063	3069	3083	3096	3108	3133
2986	2985	3006	3018	3043	3050	3056	3062	3068	3082	3095	3107	3132
2985	2984	3005	3017	3042	3049	3055	3061	3067	3081	3094	3106	3131
2984	2983	3004	3016	3041	3048	3054	3060	3066	3080	3093	3105	3130
2983	2982	3003	3015	3040	3047	3053	3059	3065	3079	3092	3104	3129
2982	2981	3002	3014	3039	3046	3052	3058	3064	3078	3091	3103	3128
2981	2980	3001	3013	3038	3045	3051	3057	3063	3077	3090	3102	3127
2980	2979	3000	3012	3037	3044	3050	3056	3062	3076	3089	3101	3126
2979	2978	2999	3011	3036	3043	3049	3055	3061	3075	3088	3100	3125
2978	2977	2998	3010	3035	3042	3048	3054	3060	3074	3087	3099	3124
2977	2976	2997	3009	3034	3041	3047	3053	3059	3073	3086	3098	3123
2976	2975	2996	3008	3033	3040	3046	3052	3058	3072	3085	3097	3122
2975	2974	2995	3007	3032	3039	3045	3051	3057	3071	3084	3096	3121
2974	2973	2994	3006	3031	3038	3044	3050	3056	3070	3083	3095	3120
2973	2972	2										

CROSS REFERENCE TABLE -- USER SYMBOLS

	13364	13405	13530	13533	13557	13563	13570	13583	13601	13604	13628	13634	13641
	13654	13666	13681	13714	13727	13753	13757	13783	13774	13784	13795	13812	13822
	13857	13974	13980	13990	13993	14027	14029	14053	13774	14073	14083	14093	14103
	14113	14125	14136	14146	14156	14174	14175	14186	14186	14204	14211	14217	14223
	14230	14231	14237	14239	14244	14259	14260	14265	14265	14289	14295	14301	14308
	14448	14481	14511	14517	14521	14528	14530	14535	14535	14558	14568	14574	14589
	14607	14613	14623	14626	14626	14637	14639	14654	14659	14714	14723	14732	14738
	14747	14758	14784	14785	14785	14797	14797	14800	14800	14866	14874	14903	14909
	14915	14921	14930	14932	14932	14937	14937	14939	14939	14994	15010	15016	15022
	15028	15034	15041	15047	15047	15051	15051	15051	15051	15051	15051	15051	15051
.DBRSH= 001400	1	1	1	1	1	1	1	1	1	1	1	1	1
	3599	3679	3740	3827	3827	4073	4190	4270	4387	4467	4584	4664	4781
	4861	4978	5058	5145	5145	5292	5550	5772	5857	5979	6064	6186	6271
	6393	6478	6500	6585	6585	6807	7014	7099	7221	7306	7372	7584	7950
	8230	8387	8500	8685	8685	8807	8929	9099	9221	9306	9372	9584	9950
.DBRSP= 003400	1	1	1	1	1	1	1	1	1	1	1	1	1
	11003	11071	11169	11188	11188	11637	11705	11954	12022	12271	12339	12588	12905
.DO = 003000	12973	13222	13290	13369	13369	13607	13607	13607	13607	13607	13607	13607	13607
	2306	2330	2330	2330	2330	2780	2897	2977	2977	3092	3104	3181	3309
	3324	3330	3330	3330	3330	3625	3625	3742	3822	3939	4019	4136	4216
	4333	4413	4530	4610	4610	4807	4807	5004	5121	5191	5301	5386	5508
	5593	5715	5800	5829	5829	6007	6129	6214	6214	6421	6543	6628	6835
	6957	7042	7164	7249	7249	7419	7419	7574	7574	7781	7896	8015	8054
	8176	8247	8333	8407	8407	8549	8549	8733	8733	8922	9095	9095	9095
	9107	9166	9167	9178	9178	9231	9231	9412	9412	9484	9484	9484	9484
	9548	9729	9730	9735	9735	9741	9800	9801	9803	9812	9855	10046	10052
	10058	10117	10118	10120	10120	10129	10182	10263	10263	10369	10375	10434	10435
	10446	10499	10580	10581	10581	10585	10585	10751	10751	10754	10763	10816	10937
	11009	11068	11069	11080	11080	11133	11133	11314	11315	11326	11365	11386	11397
	11450	11631	11632	11637	11637	11643	11702	11703	11703	11714	11767	11948	11954
	11950	12019	12020	12031	12031	12031	12094	12094	12094	12277	12336	12348	12401
	12582	12583	12594	12622	12622	12654	12655	12718	12718	12800	12911	12970	12982
	13035	13216	13217	13222	13222	13227	13287	13287	13287	13499	13552	13533	13545
	13604	13605	13616	13616	13616	13620	13621	13626	13626	14541	14613	14632	14729
	14753	14827	14836	14842	14842	14863	14906	14912	14912	14970	14976	14991	14991
.DEC = 000160	1	1	1	1	1	1	1	1	1	1	1	1	1
	10184	10250	10501	10567	10567	10818	10884	11135	11201	11452	11518	11763	11835
	12152	12403	12469	12720	12720	12786	13037	13103	13354	13420	13539	13607	13671
	14037	14398	1469	1469	1469	1469	1469	1469	1469	1469	1469	1469	1469
.DELAY 007304	1	1	1	1	1	1	1	1	1	1	1	1	1
.DMEH = 002400	1	1	1	1	1	1	1	1	1	1	1	1	1
	1562	2318	2324	2324	2324	2404	2515	2521	2595	2601	2712	2718	2792
	2909	2915	2989	2989	2989	3086	3116	3122	3175	3205	3211	3303	3342
	3348	3401	3416	3440	3440	3446	3557	3557	3637	3643	3754	3834	3840
	3951	3957	4031	4037	4037	4148	4154	4234	4234	4351	4425	4431	4542
	4548	4622	4628	4739	4739	4745	4819	4819	4942	5016	5022	5106	5121
	5165	5176	5191	5235	5235	5289	5289	5374	5442	5502	5581	5587	5703
	5709	5788	5794	5910	5910	5916	5995	6001	6117	6202	6208	6324	6330
	6409	6415	6531	6537	6537	6616	6622	6738	6744	6823	6829	6945	7030
	7036	7152	7158	7237	7237	7243	7468	7671	7711	7809	7839	7946	7994
	8054	8069	8154	8164	8164	8241	8247	8311	8321	8401	8407	8474	8856
	8875	8904	8988	8991	8991	8994	8997	9000	9003	9006	9009	9012	9018
	9021	9024	9027	9030	9030	9033	9036	9039	9042	9045	9048	9051	9057
	9060	9063	9066	9069	9069	9072	9075	9078	9081	9107	9178	9305	9311
	9314	9317	9320	9323	9323	9326	9329	9332	9335	9338	9341	9347	9350
	9353	9356	9359	9362	9362	9365	9368	9371	9374	9377	9380	9386	9389

CROSS REFERENCE TABLE -- USER SYMBOLS

.DNOP = 000000
.DOUT0= 002000
.DOUT1= 001000

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	4859	4860	4861	4862	4863	4864	4865	4866	4867	4868	4869	4870	4871	4872	4873	4874	4875	4876	4877	4878	4879	4880	4881	4882	4883	4884	4885	4886	4887	4888	4889	4890	4891	4892	4893	4894	4895	4896	4897	4898	4899	4900	4901	4902	4903	4904	4905	4906	4907	4908	4909	4910	4911	4912	4913	4914	4915	4916	4917	4918	4919	4920	4921	4922	4923	4924	4925	4926	4927	4928	4929	4930	4931	4932	4933	4934	4935	4936	4937	4938	4939	4940	4941	4942	4943	4944	4945	4946	4947	4948	4949	4950	4951	4952	4953	4954	4955	4956	4957	4958	4959	4960	4961	4962	4963	4964	4965	4966	4967	4968	4969	4970	4971	4972	4973	4974	4975	4976	4977	4978	4979	4980	4981	4982	4983	4984	4985	4986	4987	4988	4989	4990	4991	4992	4993	4994	4995	4996	4997	4998	4999	5000
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

CROSS REFERENCE TABLE -- USER SYMBOLS

.DO = 000400
.FO = 000020

.INC = 000060

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

.PLUS = 000000
.RES05 = 006122
.ROMCL = 007352
.SAV05 = 006062
.SBR = 060000

12976	13035	13038	13091	13094	13097	13100	13104	13106	13205	13219	13222	13281
13290	13333	13333	13355	13408	13411	13414	13417	13421	13423	13522	13536	13598
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	14059	14079	14089	14099
14109	14121	14131	14141	14151	14161	14171	14181	14191	14207	14220	14226	14239
14233	14235	14278	14291	14304	14317	14330	14349	14352	14390	14399	14417	14508
14533	14541	14541	14580	14583	14595	14598	14604	14623	14702	14705	14708	14711
14717	14720	14726	14729	14750	14753	14758	14771	14774	14810	14813	14818	14821
14827	14830	14836	14842	14853	14859	14866	14889	14891	14900	14906	14912	14933
14939	14955	14964	14970	14976	14988	14991						
	2297	2315	2377	2395	2494	2512	2574	2592	2691	2709	2771	2789
2688	2906	2968	2986	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					
	7830	8003	8006	8091	8094	8235	8395	8530	8614	8693	8780	8862
8910	12271	12339	13789	13965	14012	14389	14767	14809	14812	14885	14888	
	9418	9486										
	1725											
1561												
	3324	3422	7554	7611	7815	10052	10120	14541	14720	14729	14753	14827
14842	14912	14976										
	1560											
1450	1730											
1563	1559											
1436												
	2286	2294	2300	2306	2312	2337	2350	2356	2360	2374	2380	2386
2392	2417	2430	2436	2440	2483	2491	2497	2503	2509	2534	2547	2553
2557	2571	2577	2583	2589	2614	2627	2633	2637	2680	2688	2694	2700
2706	2731	2744	2750	2754	2768	2774	2780	2786	2811	2824	2830	2834
2877	2885	2891	2897	2903	2928	2941	2947	2951	2965	2971	2977	2983
3008	3021	3027	3031	3075	3083	3092	3095	3104	3110	3135	3148	3154
3158	3172	3181	3184	3193	3199	3224	3237	3243	3247	3292	3300	3309
3312	3324	3330	3336	3361	3374	3380	3384	3398	3407	3410	3422	3428
3434	3459	3472	3478	3482	3525	3533	3539	3545	3551	3576	3589	3595
3599	3613	3619	3625	3631	3656	3669	3675	3679	3722	3730	3736	3742
3748	3773	3786	3792	3796	3810	3816	3822	3828	3853	3866	3872	3876
3919	3927	3933	3939	3945	3970	3983	3989	3993	4007	4013	4019	4025
4050	4063	4069	4073	4116	4124	4130	4136	4142	4167	4180	4186	4190
4204	4210	4216	4222	4247	4260	4266	4270	4313	4321	4327	4333	4339
4364	4377	4383	4387	4401	4407	4413	4419	4444	4457	4463	4467	4510
4518	4524	4530	4536	4561	4574	4580	4584	4598	4604	4610	4616	4641
4654	4660	4664	4707	4715	4721	4727	4733	4758	4771	4777	4781	4795
4801	4807	4813	4838	4851	4857	4861	4904	4912	4918	4924	4930	4955
4968	4974	4978	4992	4998	5004	5010	5035	5048	5054	5058	5102	5115
5130	5159	5162	5165	5185	5200	5229	5232	5235	5279	5286	5289	5295
5301	5307	5310	5313	5319	5325	5358	5371	5374	5380	5386	5392	5395
5398	5404	5410	5443	5486	5453	5496	5502	5508	5514	5517	5520	5526

5532	5565	5578	5581	5587	5593	5599	5602	5605	5611	5617	5650	5693
5700	5703	5709	5715	5721	5724	5727	5733	5739	5772	5785	5788	5794
5800	5806	5809	5812	5818	5824	5857	5900	5907	5910	5916	5922	5928
5931	5934	5940	5946	5979	5992	5995	6001	6007	6013	6016	6019	6025
6031	6064	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	6342	6345	6348	6354	6360	6393	6406	6409	6415	6421	6427
6430	6433	6439	6445	6478	6501	6508	6531	6537	6543	6549	6552	6555
6561	6567	6600	6613	6616	6622	6628	6634	6637	6640	6646	6652	6685
6728	6735	6738	6744	6750	6753	6759	6762	6768	6774	6807	6820	6823
6829	6835	6841	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	7362	7368	7372	7374	7380	7386	7389	7392	7395	7398
7404	7419	7425	7428	7431	7437	7456	7462	7465	7521	7524	7527	7530
7533	7536	7539	7542	7545	7548	7554	7560	7564	7566	7572	7578	7581
7584	7587	7590	7593	7596	7599	7602	7611	7623	7626	7629	7632	7638
7656	7662	7665	7668	7676	7702	7705	7708	7764	7767	7770	7773	7776
7779	7782	7785	7788	7791	7794	7800	7806	7821	7830	7861	7907	7910
7913	7916	7919	7922	7928	7934	7946	7950	7952	7967	7970	7973	7976
7982	7997	8003	8006	8009	8021	8024	8027	8030	8045	8048	8063	8066
8085	8091	8094	8097	8103	8114	8117	8127	8130	8191	8194	8197	8207
8220	8226	8231	8236	8238	8254	8257	8259	8321	8327	8333	8339	8348
8351	8354	8354	8377	8383	8387	8395	8398	8414	8417	8459	8468	8471
8474	8483	8482	8492	8495	8501	8507	8510	8513	8542	8561	8564	8573
8582	8585	8591	8614	8617	8625	8672	8681	8684	8690	8693	8696	8702
8721	8724	8733	8743	8745	8751	8761	8774	8780	8783	8792	8838	8847
8850	8853	8853	8855	8858	8875	8901	8913	8916	8919	8970	8973	8976
8979	8982	8982	8985	8988	8991	8991	8993	8993	9151	9157	9166	9169
9187	9189	9199	9209	9212	9216	9223	9227	9233	9243	9246	9249	9302
9412	9433	9439	9445	9455	9458	9474	9483	9474	9510	9516	9526	9539
9548	9550	9554	9577	9582	9610	9613	9619	9625	9750	9756	9762	9772
9785	9791	9800	9821	9827	9833	9843	9851	9855	9867	9921	9924	9927
9930	9933	9936	10046	10067	10073	10079	10089	10102	10108	10117	10138	10144
10150	10160	10173	10182	10184	10189	10241	10244	10247	10250	10253	10363	10384
10390	10396	10406	10419	10425	10434	10455	10461	10467	10477	10490	10499	10501
10555	10558	10561	10564	10567	10570	10680	10701	10707	10713	10723	10736	10742
10751	10772	10778	10784	10794	10807	10816	10818	10872	10875	10878	10881	10884
10887	10997	11003	11018	11024	11030	11040	11053	11059	11068	11071	11089	11095
11101	11111	11124	11133	11135	11139	11152	11155	11158	11201	11204	11314	11335
11341	11347	11357	11370	11376	11386	11406	11412	11418	11428	11441	11450	11452
11506	11509	11512	11515	11518	11521	11631	11652	11658	11664	11674	11687	11693
11702	11723	11729	11735	11745	11758	11767	11769	11823	11826	11829	11832	11835
11838	11948	11969	11975	11981	11991	12004	12010	12019	12040	12046	12052	12062
12075	12084	12086	12140	12143	12146	12149	12152	12155	12240	12246	12252	12292
12298	12308	12321	12327	12336	12339	12357	12363	12369	12379	12392	12396	12403
12457	12460	12463	12466	12469	12472	12582	12588	12603	12609	12615	12625	12638
12644	12653	12656	12674	12680	12686	12696	12709	12718	12709	12714	12777	12780
12783	12786	12789	12899	12905	12920	12926	12932	12942	12955	12961	12970	12973
12991	12997	13003	13013	13026	13035	13037	13091	13094	13097	13100	13103	13106
13216	13237	13243	13249	13259	13272	13278	13287	13308	13314	13320	13330	13343
13352	13354	13408	13411	13414	13417	13420	13423	13533	13539	13554	13560	13566
13576	13589	13595	13604	13607	13625	13631	13637	13647	13660	13669	13671	13717
13720	13723	13726	13754	13789	13792	13795	13860	13863	13866	13876	13879	13882
13885	13888	13892	13895	13900	13903	13906	13909	13912	13915	13918	13921	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	14449	14453	14496	14499	14502	14505	14508	14514
14520	14536	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
1086	1558											
1	2337	2350	2356	2417	2430	2436	2534	2547	2553	2614	2627	2633
2731	2744	2750	2811	2824	2830	2928	2941	2947	3008	3021	3027	3135
3148	3194	3224	3237	3243	3361	3374	3380	3459	3472	3478	3576	3589
3595	3656	3659	3675	3773	3786	3792	3853	3866	3872	3970	3983	3989
4050	4063	4069	4167	4180	4186	4247	4260	4266	4364	4377	4383	4444
4457	4463	4461	4574	4580	4641	4654	4660	4758	4771	4777	4838	4851
4857	4955	4968	4974	5036	5048	5054	5165	5235	5249	5310	5374	5395
5496	5517	5581	5581	5703	5724	5788	5809	5910	5931	5995	6016	6117
6138	6202	6223	6241	6345	6409	6430	6531	6552	6616	6637	6738	6759
6823	6844	6945	6966	7030	7051	7153	7173	7237	7258	7389	7395	7428
7465	7581	7587	7586	7599	7623	7623	7655	7705	7867	7973	8045	8048
8063	8194	8207	8220	8226	8254	8257	8351	8364	8377	8383	8414	8417
8474	8492	8504	8588	8690	8742	8751	8774	8868	8919	9101	9138	9151
9157	9169	9209	9225	9250	9250	9474	9526	9539	9772	9785	9791	9843
9856	10089	10102	10108	10165	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	11991	12004	12010	12062	12075	12308	12321	12327
12379	12392	12625	12638	12644	12696	12709	12942	12955	12961	13013	13026	13259
13272	13278	13330	13343	13576	13589	13595	13647	13660	13795	13990	13996	14204
14217	14223	14236	14249	14276	14275	14288	14301	13660	13795	13990	13996	14204
14607	14626	14641	14660	14676	14680	14686	15001	14314	14327	14420	14435	14456
1	2286	2294	2300	2312	2321	2327	2360	2374	2380	2392	2401	2407
2440	2483	2491	2500	2512	2521	2527	2560	2571	2577	2589	2598	2604
2637	2680	2688	2694	2706	2715	2721	2754	2768	2774	2786	2795	2801
2834	2877	2885	2891	2903	2912	2918	2951	2965	2971	2983	2992	2998
3031	3075	3083	3089	3100	3110	3119	3126	3158	3172	3178	3184	3199
3208	3214	3247	3250	3262	3264	3272	3281	3326	3345	3351	3384	3398
3404	3410	3419	3430	3443	3449	3457	3461	3533	3539	3551	3560	3566
3599	3613	3619	3631	3640	3646	3679	3722	3730	3736	3748	3757	3763
3796	3810	3816	3823	3840	3843	3876	3919	3927	3933	3945	3954	3960
3993	4007	4013	4025	4034	4040	4073	4116	4124	4130	4142	4151	4157
4190	4204	4210	4225	4231	4237	4270	4313	4321	4327	4339	4348	4354
4387	4401	4407	4419	4428	4434	4467	4510	4518	4524	4536	4545	4551
4584	4598	4604	4616	4625	4631	4664	4707	4715	4721	4733	4742	4748
4781	4795	4801	4813	4825	4831	4864	4907	4912	4918	4930	4939	4945
4978	4992	4998	5010	5019	5025	5058	5102	5109	5112	5115	5127	5130
5156	5179	5182	5185	5197	5200	5236	5279	5286	5292	5295	5307	5313
5319	5325	5336	5348	5354	5358	5377	5377	5380	5392	5398	5404	5410
5420	5433	5439	5443	5448	5453	5499	5499	5514	5520	5526	5532	5542
5555	5561	5565	5578	5584	5587	5593	5605	5611	5617	5627	5648	5648
5650	5693	5708	5706	5709	5721	5727	5733	5739	5749	5762	5768	5772
5785	5791	5794	5806	5812	5818	5824	5834	5847	5853	5857	5900	5907
5913	5916	5928	5934	5940	5946	5966	5969	5975	5979	5992	5998	6001
6013	6019	6025	6031	6041	6054	6060	6064	6107	6114	6120	6123	6135
6141	6147	6153	6163	6176	6182	6186	6199	6205	6208	6220	6226	6232
6238	6248	6261	6267	6271	6314	6321	6327	6330	6342	6348	6354	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	6756	6762	6768	6774	6784	6797	6803	6807	6820	6826
6829	6841	6847	6853	6859	6868	6883	6888	6892	6903	6912	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
7382	7392	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	7622	7632	7638	7643	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	8066	8072	8142	8157	8160	8164	8167
8170	8182	8185	8191	8197	8230	8299	8314	8317	8321	8324	8327	8339
8342	8348	8354	8367	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	9504	9507	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	10663	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	11506	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	12268	12283	12286	12292	12298	12300	12342	12354	12357
12363	12369	12457	12460	12463	12466	12472	12585	12600	12603	12609	12615	12647
12659	12671	12674	12680	12686	12774	12777	12780	12783	12789	12902	12917	12920
12926	12932	12934	12976	12988	12991	12997	13003	13091	13094	13097	13100	13106
13219	13234	13237	13243	13249	13281	13293	13305	13308	13314	13320	13408	13411
13414	13417	13423	13426	13431	13454	13460	13466	13498	13510	13522	13525	13631
13637	13717	13720	13723	13726	13738	13754	13770	13808	13860	13863	13866	13873
13879	13882	13886	13889	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	14355	14367	14370	14376	14417	14432	14438	14444	14490	14493	14496
14499	14502	14506	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	3168	3175	3205	3211	3218	3231	3254	3289	3296
3303	3318	3342	3348	3355	3368	3394	3401	3416	3440	3446	3453	3466
3489	3522	3529	3557	3563	3570	3583	3609	3637	3643	3650	3663	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

4555	4568	4594	4622	4646	4648	4671	4704	4711	4739	4745	4752
4765	4798	4819	4829	4835	4868	4901	4908	4936	4942	4949	4962
4988	5016	5022	5029	5035	5065	5106	5133	5137	5147	5176	5203
5217	5248	5253	5276	5277	5281	5306	5343	5346	5401	5407	5414
5427	5450	5497	5490	5517	5524	5575	5575	5588	55614	5621	5634
5657	5690	5697	5730	5730	5736	5815	5828	5821	5828	5841	5854
5897	5904	5937	5943	5943	5956	6028	6028	6035	6048	6071	6084
6111	6144	6150	6157	6170	6196	6242	6242	6255	6278	6311	6318
6351	6374	6364	6377	6403	6436	6463	6463	6485	6518	6525	6558
6564	6571	6584	6610	6643	6649	6682	6682	6725	6732	6765	6771
6778	6791	6817	6850	6856	6862	6893	6893	6929	6972	6978	6985
6998	7024	7057	7063	7076	7083	7106	7106	7179	7185	7192	7205
7231	7264	7270	7277	7290	7313	7344	7344	7383	7401	7434	7441
7468	7475	7485	7518	7557	7569	7575	7575	7671	7678	7711	7718
7728	7761	7797	7803	7809	7818	7839	7839	7865	7898	7901	7904
7925	7931	7937	7940	7943	7961	7964	7964	8015	8033	8036	8039
8069	8076	8104	8139	8145	8148	8151	8151	8201	8214	8241	8254
8246	8302	8305	8308	8311	8314	8328	8328	8401	8415	8415	8455
8498	8511	8521	8536	8552	8555	8558	8558	8605	8620	8637	8669
8675	8678	8687	8712	8715	8718	8748	8748	8796	8803	8835	8841
8844	8856	8875	8882	8882	8884	8926	8926	8967	8983	8995	8994
8997	9000	9003	9006	9009	9012	9015	9015	9024	9027	9030	9033
9036	9039	9042	9045	9048	9051	9054	9054	9063	9066	9069	9072
9075	9078	9081	9084	9088	9092	9119	9119	9145	9163	9190	9196
9203	9216	9228	9243	9278	9281	9294	9294	9311	9314	9317	9320
9323	9326	9329	9332	9335	9338	9341	9341	9359	9363	9366	9369
9362	9365	9368	9371	9374	9377	9380	9380	9399	9403	9406	9409
9401	9405	9409	9436	9442	9449	9463	9463	9513	9520	9533	9545
9560	9575	9598	9601	9622	9625	9628	9628	9637	9640	9643	9646
9649	9652	9655	9658	9661	9664	9667	9667	9676	9679	9682	9685
9688	9691	9694	9697	9700	9703	9706	9706	9715	9718	9722	9726
9753	9759	9766	9779	9797	9797	9830	9830	9832	9877	9912	9915
9918	9939	9942	9945	9948	9951	9954	9954	9963	9966	9969	9972
9975	9978	9981	9984	9987	9990	9993	9993	10002	10005	10008	10011
10014	10017	10020	10023	10026	10029	10032	10032	10043	10046	10049	10052
10096	10114	10141	10147	10154	10167	10176	10176	10189	10192	10195	10198
10262	10265	10268	10271	10274	10277	10280	10280	10293	10296	10299	10302
10301	10304	10307	10310	10313	10316	10319	10319	10332	10335	10338	10341
10340	10343	10346	10349	10352	10355	10358	10358	10371	10374	10377	10380
10464	10471	10484	10496	10511	10546	10549	10549	10576	10579	10582	10585
10588	10591	10594	10597	10600	10603	10606	10606	10615	10618	10621	10624
10627	10630	10633	10636	10639	10642	10645	10645	10654	10657	10660	10663
10666	10669	10673	10677	10680	10683	10686	10686	10695	10698	10701	10704
10813	10828	10863	10866	10874	10890	10893	10893	10902	10905	10908	10911
10914	10917	10920	10923	10926	10929	10932	10932	10941	10944	10947	10950
10953	10956	10959	10962	10965	10968	10971	10971	10980	10983	10986	10989
10994	11021	11027	11034	11047	11055	11092	11092	11118	11130	11145	11180
11183	11186	11207	11210	11213	11216	11219	11219	11228	11231	11234	11237
11240	11243	11246	11249	11252	11255	11258	11258	11267	11270	11273	11276
11279	11282	11285	11288	11291	11294	11297	11297	11307	11311	11318	11344
11351	11364	11382	11409	11415	11422	11435	11435	11447	11500	11503	11524
11527	11530	11533	11536	11539	11542	11545	11545	11551	11557	11560	11563
11566	11569	11572	11575	11578	11581	11584	11584	11593	11596	11599	11602
11605	11608	11611	11614	11617	11620	11624	11624	11631	11668	11681	11699
11726	11732	11739	11752	11764	11779	11814	11814	11817	11844	11847	11850
11853	11856	11859	11862	11865	11868	11871	11871	11874	11880	11886	11889

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	11985	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	12475	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	13022	13024	13026	13028	13030	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	13209	13213	13240	13246	13253	13266	13284	13311	13317
	13324	13337	13349	13364	13379	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	13537	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	14195	14211	14211	14211	14256	14269	14282
	14295	14308	14321	14334	14343	14346	14352	14373	14380	14380	14441	14448	14481
	14484	14487	14511	14517	14523	14523	14538	14550	14562	14562	14654	14654	14699
	14714	14723	14747	14821	14833	14839	14866	14894	14903	14903	14936	14958	14967
	14973	14985	15010	15016	15022	15041							
.SINO = 020000	1	3536	3554	3616	3634	3733	3751	3813	3831		3948	4010	4028
	4127	4145	4207	4225	4324	4342	4404	4422	4521	4539	4601	4619	4718
	4736	4798	4816	4915	4933	4995	5013	7413	7450	7647	7650	7687	7690
.SINI = 120000	7991	7994	14121	14131	14141	14151	14161	14171	14181	14191	14604	14623	2789
	2888	2906	2968	2986	3088	3113	3187	3202	3215	3239	2709	2771	7407
	7551	7608	7614	7641	7812	7824	7858	7895	8215	8215	3413	3437	7407
	14069	14079	14089	14099	14109	14532	14544	14556	14568	14574	14580	14583	14589
	14595	14598	14717	14726	14732	14738	14750	14758	14784	14824	14830	14845	14851
	14900	14915	14921	14964	14979	14988	14994	15028	15034	15047			
.SPEM = 040000	1	2321	2327	2401	2407	2518	2524	2538	2604	2715	2721	2795	2801
	2912	2918	2992	2998	3089	3119	3125	3178	3208	3214	3306	3321	3345
	3351	3404	3419	3443	3449	3560	3566	3640	3646	3757	3763	3837	3843
	3954	3960	4034	4040	4151	4157	4231	4237	4348	4354	4428	4434	4545
	4551	4625	4631	4742	4748	4822	4828	4939	4945	5019	5025	5109	5112
	5121	5127	5156	5179	5182	5191	5197	5258	5268	5336	5348	5354	5377
	5420	5433	5439	5499	5542	5555	5561	5584	5629	5636	5646	5706	5749
	5762	5768	5791	5834	5847	5863	5913	5956	5969	5975	5998	6041	6054
	6060	6120	6163	6176	6182	6205	6248	6261	6267	6327	6370	6383	6389
	6412	6455	6468	6474	6534	6577	6590	6596	6619	6662	6675	6681	6741
	6784	6797	6803	6826	6869	6882	6888	6948	6991	7004	7010	7033	7076
	7089	7095	7155	7188	7211	7217	7240	7283	7296	7302	7471	7674	7714
	7815	7842	8042	8054	8060	8072	8157	8160	8167	8185	8247	8314	8317
	8324	8342	8407	8477	8489	8567	8579	8727	8739	8859	8878	8907	8998
	9107	9113	9160	9172	9178	9184	9415	9418	9424	9430	9477	9486	9489
	9495	9501	9732	9735	9741	9747	9794	9803	9806	9812	9818	10049	10052
	10058	10064	10111	10120	10123	10129	10135	10365	10369	10375	10381	10428	10437
	10440	10446	10452	10683	10686	10692	10692	10745	10754	10757	10763	10769	11000

CROSS REFERENCE TABLE -- USER SYMBOLS

.START 002402
.SUBMC= 000040
.SUB1C= 000340
.SUB2C= 000360

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

11009	11016	11063	11074	11080	11086	11317	11320	11326	11332	11379	11388	11391
11197	11198	11139	11137	11143	11149	11156	11170	11178	11174	11170	11151	11154
11160	11162	11013	11022	11025	11031	11037	11268	11277	11283	11230	11242	11248
12051	12052	12000	12000	12017	12059	12065	12071	12902	12911	12917	12964	12976
12082	12083	12074	12079	12028	12034	12081	12090	12093	12044	12005	12036	12045
13551	13550	13610	13616	13622	13738	13770	13808	14355	14367	14526	13536	13545
233	732	748	1828									
	11954	12023	8702	8792	11320	11388						
	11954	12023	2503	2583	2700	2780	2897	2977	3104	3193	3330	3428
	3545	4024	3823	3939	4019	4136	4216	4333	4413	4530	4610	4727
	4807	4824	5004	5121	5301	5386	5508	5593	5715	5800	5922	6007
	6129	6214	6336	6421	6543	6628	6825	6957	7042	7164	7249	7419
	7456	7625	7696	8054	8176	8247	8333	8483	8573	8733	9107	9178
	9424	9495	9741	9812	10058	10129	10375	10688	10763	11009	11080	11326
	11397	11643	11714	11960	12031	12277	12348	12594	12665	12911	13222	13228
	13290	13299	13545	13616	13980	14361	14426	14613	14632			
	1565	1752	10437									
	1178	1181										
	486	491										
	2293	2294	2286	2287	2290	2291	2294	2295	2297	2298	2300	2306
	2307	2309	2310	2312	2313	2315	2316	2318	2319	2321	2322	2325
	2327	2328	2331	2332	2334	2335	2337	2338	2340	2341	2344	2347
	2348	2349	2351	2352	2354	2355	2357	2358	2360	2361	2365	2367
	2374	2375	2377	2378	2380	2381	2383	2384	2386	2389	2390	2395
	2396	2397	2399	2401	2402	2404	2405	2407	2408	2410	2411	2415
	2417	2418	2420	2421	2422	2424	2425	2427	2428	2430	2431	2436
	2437	2438	2440	2441	2442	2444	2445	2447	2448	2450	2451	2454
	2457	2458	2460	2461	2462	2464	2465	2467	2468	2470	2471	2474
	2487	2488	2490	2491	2492	2494	2495	2497	2498	2500	2501	2504
	2510	2511	2513	2514	2516	2517	2518	2520	2521	2522	2523	2526
	2531	2532	2534	2535	2537	2538	2540	2541	2542	2544	2545	2548
	2557	2558	2560	2561	2562	2564	2565	2567	2568	2570	2571	2574
	2587	2588	2590	2591	2592	2594	2595	2597	2598	2600	2601	2604
	2610	2611	2613	2614	2616	2617	2618	2620	2621	2622	2623	2626
	2637	2638	2640	2641	2642	2644	2645	2647	2648	2650	2651	2654
	2667	2668	2670	2671	2672	2674	2675	2676	2678	2679	2680	2683
	2691	2692	2694	2695	2696	2698	2699	2700	2702	2703	2704	2707
	2713	2714	2716	2717	2718	2719	2720	2721	2722	2723	2724	2726
	2734	2735	2736	2737	2738	2739	2740	2741	2742	2743	2744	2746
	2757	2758	2760	2761	2762	2763	2764	2765	2766	2767	2768	2770
	2783	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2795
	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2813	2814
	2824	2825	2827	2828	2829	2830	2831	2832	2833	2834	2835	2836
	2848	2849	2851	2852	2853	2854	2855	2856	2857	2858	2859	2860
	2897	2898	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909
	2916	2918	2919	2920	2921	2922	2923	2924	2925	2926	2927	2928
	2938	2939	2941	2942	2943	2944	2945	2946	2947	2948	2949	2950
	2962	2963	2965	2966	2967	2968	2969	2970	2971	2972	2973	2974
	2986	2987	2989	2990	2991	2992	2993	2994	2995	2996	2997	2998
	3006	3008	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018
	3027	3028	3030	3031	3032	3033	3034	3035	3036	3037	3038	3039
	3076	3079	3080	3081	3082	3083	3084	3085	3086	3087	3088	3089
	3098	3099	3104	3105	3106	3107	3108	3109	3110	3111	3112	3113
	3120	3122	3123	3125	3126	3129	3130	3132	3133	3134	3135	3136

CROSS REFERENCE TABLE -- USER SYMBOLS

3142	3143	3145	3146	3148	3149	3151	3152	3154	3155	3160	3161	3163
3164	3168	3169	3172	3173	3175	3176	3178	3179	3181	3182	3184	3185
3187	3188	3193	3194	3196	3197	3199	3200	3202	3203	3205	3206	3208
3209	3211	3212	3214	3215	3218	3219	3221	3222	3223	3225	3227	3228
3231	3232	3233	3235	3237	3238	3240	3241	3243	3244	3247	3250	3254
3255	3257	3258	3260	3261	3263	3265	3266	3267	3268	3270	3271	3274
3275	3276	3277	3279	3281	3282	3283	3284	3285	3286	3287	3288	3291
3292	3293	3294	3295	3296	3297	3298	3299	3300	3301	3302	3303	3304
3305	3306	3307	3308	3309	3310	3312	3313	3315	3316	3318	3319	3321
3322	3323	3324	3325	3326	3327	3328	3329	3330	3331	3332	3333	3334
3335	3336	3337	3338	3339	3340	3341	3342	3343	3344	3345	3346	3347
3348	3349	3350	3351	3352	3353	3354	3355	3356	3357	3358	3359	3360
3361	3362	3363	3364	3365	3366	3367	3368	3369	3370	3371	3372	3373
3374	3375	3376	3377	3378	3379	3380	3381	3382	3383	3384	3385	3386
3387	3388	3389	3390	3391	3392	3393	3394	3395	3396	3397	3398	3399
3400	3401	3402	3403	3404	3405	3406	3407	3408	3409	3410	3411	3412
3413	3414	3415	3416	3417	3418	3419	3420	3421	3422	3423	3424	3425
3426	3427	3428	3429	3430	3431	3432	3433	3434	3435	3436	3437	3438
3439	3440	3441	3442	3443	3444	3445	3446	3447	3448	3449	3450	3451
3452	3453	3454	3455	3456	3457	3458	3459	3460	3461	3462	3463	3464
3465	3466	3467	3468	3469	3470	3471	3472	3473	3474	3475	3476	3477
3478	3479	3480	3481	3482	3483	3484	3485	3486	3487	3488	3489	3490
3491	3492	3493	3494	3495	3496	3497	3498	3499	3500	3501	3502	3503
3504	3505	3506	3507	3508	3509	3510	3511	3512	3513	3514	3515	3516
3517	3518	3519	3520	3521	3522	3523	3524	3525	3526	3527	3528	3529
3530	3531	3532	3533	3534	3535	3536	3537	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									

CROSS REFERENCE TABLE -- UBER SYMBOLS

4574	4575	4576	4577	4578	4579	4580	4581	4586
4587	4588	4589	4590	4591	4592	4593	4594	4595
4613	4614	4615	4616	4617	4618	4619	4620	4621
4630	4631	4632	4633	4634	4635	4636	4637	4638
4654	4655	4656	4657	4658	4659	4660	4661	4662
4678	4679	4680	4681	4682	4683	4684	4685	4686
4727	4728	4729	4730	4731	4732	4733	4734	4735
4746	4747	4748	4749	4750	4751	4752	4753	4754
4768	4769	4770	4771	4772	4773	4774	4775	4776
4792	4793	4794	4795	4796	4797	4798	4799	4800
4816	4817	4818	4819	4820	4821	4822	4823	4824
4836	4837	4838	4839	4840	4841	4842	4843	4844
4857	4858	4859	4860	4861	4862	4863	4864	4865
4905	4906	4907	4908	4909	4910	4911	4912	4913
4930	4931	4932	4933	4934	4935	4936	4937	4938
4950	4951	4952	4953	4954	4955	4956	4957	4958
4971	4972	4973	4974	4975	4976	4977	4978	4979
4996	4997	4998	4999	5000	5001	5002	5003	5004
5019	5020	5021	5022	5023	5024	5025	5026	5027
5039	5040	5041	5042	5043	5044	5045	5046	5047
5055	5056	5057	5058	5059	5060	5061	5062	5063
5081	5082	5083	5084	5085	5086	5087	5088	5089
5109	5110	5111	5112	5113	5114	5115	5116	5117
5133	5134	5135	5136	5137	5138	5139	5140	5141
5154	5155	5156	5157	5158	5159	5160	5161	5162
5176	5177	5178	5179	5180	5181	5182	5183	5184
5198	5199	5200	5201	5202	5203	5204	5205	5206
5221	5222	5223	5224	5225	5226	5227	5228	5229
5247	5248	5249	5250	5251	5252	5253	5254	5255
5271	5272	5273	5274	5275	5276	5277	5278	5279
5297	5298	5299	5300	5301	5302	5303	5304	5305
5321	5322	5323	5324	5325	5326	5327	5328	5329
5347	5348	5349	5350	5351	5352	5353	5354	5355
5371	5372	5373	5374	5375	5376	5377	5378	5379
5397	5398	5399	5400	5401	5402	5403	5404	5405
5421	5422	5423	5424	5425	5426	5427	5428	5429
5447	5448	5449	5450	5451	5452	5453	5454	5455
5471	5472	5473	5474	5475	5476	5477	5478	5479
5497	5498	5499	5500	5501	5502	5503	5504	5505
5521	5522	5523	5524	5525	5526	5527	5528	5529
5547	5548	5549	5550	5551	5552	5553	5554	5555
5571	5572	5573	5574	5575	5576	5577	5578	5579
5597	5598	5599	5600	5601	5602	5603	5604	5605
5621	5622	5623	5624	5625	5626	5627	5628	5629
5647	5648	5649	5650	5651	5652	5653	5654	5655
5671	5672	5673	5674	5675	5676	5677	5678	5679
5697	5698	5699	5700	5701	5702	5703	5704	5705
5721	5722	5723	5724	5725	5726	5727	5728	5729
5747	5748	5749	5750	5751	5752	5753	5754	5755
5771	5772	5773	5774	5775	5776	5777	5778	5779
5797	5798	5799	5800	5801	5802	5803	5804	5805
5821	5822	5823	5824	5825	5826	5827	5828	5829
5847	5848	5849	5850	5851	5852	5853	5854	5855
5871	5872	5873	5874	5875	5876	5877	5878	5879
5897	5898	5899	5900	5901	5902	5903	5904	5905
5921	5922	5923	5924	5925	5926	5927	5928	5929
5947	5948	5949	5950	5951	5952	5953	5954	5955
5971	5972	5973	5974	5975	5976	5977	5978	5979
5997	5998	5999	6000	6001	6002	6003	6004	6005

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	6052	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	6226	6227	6229	6230	6232	6233	6236	6238	6239
6239	6240	6243	6245	6246	6246	6249	6251	6252	6253	6255	6258	6259
6261	6262	6263	6265	6267	6268	6273	6274	6278	6279	6281	6282	6284
6285	6286	6287	6289	6291	6292	6293	6294	6295	6297	6298	6299	6300
6300	6301	6303	6307	6309	6310	6311	6312	6313	6314	6315	6317	6318
6319	6320	6321	6322	6323	6324	6325	6326	6327	6328	6329	6330	6331
6332	6333	6334	6335	6336	6337	6338	6339	6340	6341	6342	6343	6344
6345	6346	6347	6348	6349	6350	6351	6352	6353	6354	6355	6356	6357
6358	6359	6360	6361	6362	6363	6364	6365	6366	6367	6368	6369	6370
6371	6372	6373	6374	6375	6376	6377	6378	6379	6380	6381	6382	6383
6384	6385	6386	6387	6388	6389	6390	6391	6392	6393	6394	6395	6396
6397	6398	6399	6400	6401	6402	6403	6404	6405	6406	6407	6408	6409
6410	6411	6412	6413	6414	6415	6416	6417	6418	6419	6420	6421	6422
6423	6424	6425	6426	6427	6428	6429	6430	6431	6432	6433	6434	6435
6436	6437	6438	6439	6440	6441	6442	6443	6444	6445	6446	6447	6448
6449	6450	6451	6452	6453	6454	6455	6456	6457	6458	6459	6460	6461
6462	6463	6464	6465	6466	6467	6468	6469	6470	6471	6472	6473	6474
6475	6476	6477	6478	6479	6480	6481	6482	6483	6484	6485	6486	6487
6488	6489	6490	6491	6492	6493	6494	6495	6496	6497	6498	6499	6500
6501	6502	6503	6504	6505	6506	6507	6508	6509	6510	6511	6512	6513
6514	6515	6516	6517	6518	6519	6520	6521	6522	6523	6524	6525	6526
6527	6528	6529	6530	6531	6532	6533	6534	6535	6536	6537	6538	6539
6540	6541	6542	6543	6544	6545	6546	6547	6548	6549	6550	6551	6552
6553	6554	6555	6556	6557	6558	6559	6560	6561	6562	6563	6564	6565
6566	6567	6568	6569	6570	6571	6572	6573	6574	6575	6576	6577	6578
6579	6580	6581	6582	6583	6584	6585	6586	6587	6588	6589	6590	6591
6592	6593	6594	6595	6596	6597	6598	6599	6600	6601	6602	6603	6604
6605	6606	6607	6608	6609	6610	6611	6612	6613	6614	6615	6616	6617
6618	6619	6620	6621	6622	6623	6624	6625	6626	6627	6628	6629	6630
6631	6632	6633	6634	6635	6636	6637	6638	6639	6640	6641	6642	6643
6644	6645	6646	6647	6648	6649	6650	6651	6652	6653	6654	6655	6656
6657	6658	6659	6660	6661	6662	6663	6664	6665	6666	6667	6668	6669
6670	6671	6672	6673	6674	6675	6676	6677	6678	6679	6680	6681	6682
6683	6684	6685	6686	6687	6688	6689	6690	6691	6692	6693	6694	6695
6696	6697	6698	6699	6700	6701	6702	6703	6704	6705	6706	6707	6708
6709	6710	6711	6712	6713	6714	6715	6716	6717	6718	6719	6720	6721
6722	6723	6724	6725	6726	6727	6728	6729	6730	6731	6732	6733	6734
6735	6736	6737	6738	6739	6740	6741	6742	6743	6744	6745	6746	6747
6748	6749	6750	6751	6752	6753	6754	6755	6756	6757	6758	6759	6760
6761	6762	6763	6764	6765	6766	6767	6768	6769	6770	6771	6772	6773
6774	6775	6776	6777	6778	6779	6780	6781	6782	6783	6784	6785	6786
6787	6788	6789	6790	6791	6792	6793	6794	6795	6796	6797	6798	6799
6800	6801	6802	6803	6804	6805	6806	6807	6808	6809	6810	6811	6812
6813	6814	6815	6816	6817	6818	6819	6820	6821	6822	6823	6824	6825
6826	6827	6828	6829	6830	6831	6832	6833	6834	6835	6836	6837	6838
6839	6840	6841	6842	6843	6844	6845	6846	6847	6848	6849	6850	6851
6852	6853	6854	6855	6856	6857	6858	6859	6860	6861	6862	6863	6864
6865	6866	6867	6868	6869	6870	6871	6872	6873	6874	6875	6876	6877
6878	6879	6880	6881	6882	6883	6884	6885	6886	6887	6888	6889	6890
6891	6892	6893	6894	6895	6896	6897	6898	6899	6900	6901	6902	6903
6904	6905	6906	6907	6908	6909	6910	6911	6912	6913	6914	6915	6916
6917	6918	6919	6920	6921	6922	6923	6924	6925	6926	6927	6928	6929
6930	6931	6932	6933	6934	6935	6936	6937	6938	6939	6940	6941	6942
6943	6944	6945	6946	6947	6948	6949	6950	6951	6952	6953	6954	6955
6956	6957	6958	6959	6960	6961	6962	6963	6964	6965	6966	6967	6968
6969	6970	6971	6972	6973	6974	6975	6976	6977	6978	6979	6980	6981
6982	6983	6984	6985	6986	6987	6988	6989	6990	6991	6992	6993	6994
6995	6996	6997	6998	6999	7000	7001	7002	7003	7004	7005	7006	7007
7008	7009	7010	7011	7012	7013	7014	7015	7016	7017	7018	7019	7020
7021	7022	7023	7024	7025	7026	7027	7028	7029	7030	7031	7032	7033
7034	7035	7036	7037	7038	7039	7040	7041	7042	7043	7044	7045	7046
7047	7048	7049	7050	7051	7052	7053	7054	7055	7056	7057	7058	7059
7060	7061	7062	7063	7064	7065	7066	7067	7068	7069	7070	7071	7072
7073	7074	7075	7076	7077	7078	7079	7080	7081	7082	7083	7084	7085
7086	7087	7088	7089	7090	7091	7092	7093	7094	7095	7096	7097	7098
7099	7100	7101	7102	7103	7104	7105	7106	7107	7108	7109	7110	7111
7112	7113	7114	7115	7116	7117	7118	7119	7120	7121	7122	7123	7124
7125	7126	7127	7128	7129	7130	7131	7132	7133	7134	7135	7136	7137
7138	7139	7140	7141	7142	7143	7144	7145	7146	7147	7148	7149	7150
7151	7152	7153	7154	7155	7156	7157	7158	7159	7160	7161	7162	7163
7164	7165	7166	7167	7168	7169	7170	7171	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	7201	7202
7203	7204	7205	7206	7207	7208	7209	7210	7211	7212	7213	7214	7215
7216	7217	7218	7219	7220	7221	7222	7223	7224	7225	7226	7227	7228
7229	7230	7231	7232	7233	7234	7235	7236	7237	7238	7239	7240	7241
7242	7243	7244	7245	7246	7247	7248	7249	7250	7251	7252	7253	7254
7255	7256	7257	7258	7259	7260	7261	7262	7263	7264	7265	7266	7267
7268	7269	7270	7271	7272	7273	7274	7275	7276	7277	7278	7279	7280
7281	7282	7283	7284	7285	7286	7287	7288	7289	7290	7291	7292	7293
7294	7295	7296	7297	7298	7299	7300	7301	7302	7303	7304	7305	7306
7307	7308	7309	7310	7311	7312	7313	7314	7315	7316	7317	7318	7319
7320	7321	7322	7323	7324	7325	7326	7327	7328	7329	7330	7331	7332
7333	7334	7335	7336	7337	7338	7339	7340	7341	7342	7343	7344	7345
7346	7347	7348	7349	7350	7351	7352	7353	7354	7355	7356	7357	7358
7359	7360	7361	7362	7363	7364	7365	7366	7367	7368	7369	7370	7371

CROSS REFERENCE TABLE -- USER SYMBOLS

7374	7375	7377	7378	7380	7381	7383	7384	7386	7387	7389	7390	7392
7393	7395	7396	7397	7398	7399	7400	7401	7402	7403	7404	7405	7406
7407	7408	7409	7410	7411	7412	7413	7414	7415	7416	7417	7418	7419
7420	7421	7422	7423	7424	7425	7426	7427	7428	7429	7430	7431	7432
7433	7434	7435	7436	7437	7438	7439	7440	7441	7442	7443	7444	7445
7446	7447	7448	7449	7450	7451	7452	7453	7454	7455	7456	7457	7458
7459	7460	7461	7462	7463	7464	7465	7466	7467	7468	7469	7470	7471
7472	7473	7474	7475	7476	7477	7478	7479	7480	7481	7482	7483	7484
7485	7486	7487	7488	7489	7490	7491	7492	7493	7494	7495	7496	7497
7498	7499	7500	7501	7502	7503	7504	7505	7506	7507	7508	7509	7510
7511	7512	7513	7514	7515	7516	7517	7518	7519	7520	7521	7522	7523
7524	7525	7526	7527	7528	7529	7530	7531	7532	7533	7534	7535	7536
7537	7538	7539	7540	7541	7542	7543	7544	7545	7546	7547	7548	7549
7550	7551	7552	7553	7554	7555	7556	7557	7558	7559	7560	7561	7562
7563	7564	7565	7566	7567	7568	7569	7570	7571	7572	7573	7574	7575
7576	7577	7578	7579	7580	7581	7582	7583	7584	7585	7586	7587	7588
7589	7590	7591	7592	7593	7594	7595	7596	7597	7598	7599	7600	7601
7602	7603	7604	7605	7606	7607	7608	7609	7610	7611	7612	7613	7614
7615	7616	7617	7618	7619	7620	7621	7622	7623	7624	7625	7626	7627
7628	7629	7630	7631	7632	7633	7634	7635	7636	7637	7638	7639	7640
7641	7642	7643	7644	7645	7646	7647	7648	7649	7650	7651	7652	7653
7654	7655	7656	7657	7658	7659	7660	7661	7662	7663	7664	7665	7666
7667	7668	7669	7670	7671	7672	7673	7674	7675	7676	7677	7678	7679
7680	7681	7682	7683	7684	7685	7686	7687	7688	7689	7690	7691	7692
7693	7694	7695	7696	7697	7698	7699	7700	7701	7702	7703	7704	7705
7706	7707	7708	7709	7710	7711	7712	7713	7714	7715	7716	7717	7718
7719	7720	7721	7722	7723	7724	7725	7726	7727	7728	7729	7730	7731
7732	7733	7734	7735	7736	7737	7738	7739	7740	7741	7742	7743	7744
7745	7746	7747	7748	7749	7750	7751	7752	7753	7754	7755	7756	7757
7758	7759	7760	7761	7762	7763	7764	7765	7766	7767	7768	7769	7770
7771	7772	7773	7774	7775	7776	7777	7778	7779	7780	7781	7782	7783
7784	7785	7786	7787	7788	7789	7790	7791	7792	7793	7794	7795	7796
7797	7798	7799	7800	7801	7802	7803	7804	7805	7806	7807	7808	7809
7810	7811	7812	7813	7814	7815	7816	7817	7818	7819	7820	7821	7822
7823	7824	7825	7826	7827	7828	7829	7830	7831	7832	7833	7834	7835
7836	7837	7838	7839	7840	7841	7842	7843	7844	7845	7846	7847	7848
7849	7850	7851	7852	7853	7854	7855	7856	7857	7858	7859	7860	7861
7862	7863	7864	7865	7866	7867	7868	7869	7870	7871	7872	7873	7874
7875	7876	7877	7878	7879	7880	7881	7882	7883	7884	7885	7886	7887
7888	7889	7890	7891	7892	7893	7894	7895	7896	7897	7898	7899	7900
7901	7902	7903	7904	7905	7906	7907	7908	7909	7910	7911	7912	7913
7914	7915	7916	7917	7918	7919	7920	7921	7922	7923	7924	7925	7926
7927	7928	7929	7930	7931	7932	7933	7934	7935	7936	7937	7938	7939
7940	7941	7942	7943	7944	7945	7946	7947	7948	7949	7950	7951	7952
7953	7954	7955	7956	7957	7958	7959	7960	7961	7962	7963	7964	7965
7966	7967	7968	7969	7970	7971	7972	7973	7974	7975	7976	7977	7978
7979	7980	7981	7982	7983	7984	7985	7986	7987	7988	7989	7990	7991
7992	7993	7994	7995	7996	7997	7998	7999	8000	8001	8002	8003	8004
8005	8006	8007	8008	8009	8010	8011	8012	8013	8014	8015	8016	8017
8018	8019	8020	8021	8022	8023	8024	8025	8026	8027	8028	8029	8030
8031	8032	8033	8034	8035	8036	8037	8038	8039	8040	8041	8042	8043
8044	8045	8046	8047	8048	8049	8050	8051	8052	8053	8054	8055	8056
8057	8058	8059	8060	8061	8062	8063	8064	8065	8066	8067	8068	8069
8070	8071	8072	8073	8074	8075	8076	8077	8078	8079	8080	8081	8082
8083	8084	8085	8086	8087	8088	8089	8090	8091	8092	8093	8094	8095
8096	8097	8098	8099	8100	8101	8102	8103	8104	8105	8106	8107	8108
8109	8110	8111	8112	8113	8114	8115	8116	8117	8118	8119	8120	8121
8122	8123	8124	8125	8126	8127	8128	8129	8130	8131	8132	8133	8134
8135	8136	8137	8138	8139	8140	8141	8142	8143	8144	8145	8146	8147
8148	8149	8150	8151	8152	8153	8154	8155	8156	8157	8158	8159	8160
8161	8162	8163	8164	8165	8166	8167	8168	8169	8170	8171	8172	8173
8174	8175	8176	8177	8178	8179	8180	8181	8182	8183	8184	8185	8186
8187	8188	8189	8190	8191	8192	8193	8194	8195	8196	8197	8198	8199
8200	8201	8202	8203	8204	8205	8206	8207	8208	8209	8210	8211	8212
8213	8214	8215	8216	8217	8218	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	8568	8569	8570	8571	8572	8573	8574	8575	8576
8577	8578	8579	8580	8581	8582	8583	8584	8585	8586	8587	8588	8589
8590	8591	8592	8593	8594	8595	8596	8597	8598	8599	8600	8601	8602
8603	8604	8605	8606	8607	8608	8609	8610	8611	8612	8613	8614	8615
8616	8617	8618	8619	8620	8621	8622	8623	8624	8625	8626	8627	8628
8629	8630	8631	8632	8633	8634	8635	8636	8637	8638	8639	8640	8641
8642	8643	8644	8645	8646	8647	8648	8649	8650	8651	8652	8653	8654
8655	8656	8657	8658	8659	8660	8661	8662	8663	8664	8665	8666	8667
8668	8669	8670	8671	8672	8673							

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	8776	8778	8780
8781	8783	8784	8786	8787	8789	8793	8795	8796	8798	8799	8803	8804
8806	8807	8809	8810	8815	8816	8817	8819	8820	8822	8824	8825	8827
8848	8850	8851	8853	8854	8856	8857	8859	8860	8862	8863	8865	8866
8889	8892	8893	8895	8896	8898	8899	8901	8902	8903	8905	8907	8908
8910	8911	8913	8914	8916	8917	8919	8920	8921	8922	8923	8927	8929
8930	8932	8933	8936	8937	8938	8940	8941	8942	8943	8944	8947	8948
8973	8977	8979	8980	8981	8983	8984	8985	8986	8987	8989	8990	8994
8995	8997	8998	9000	9001	9003	9004	9005	9007	9008	9010	9012	9013
9015	9016	9018	9019	9021	9022	9024	9025	9027	9028	9029	9031	9033
9034	9036	9037	9039	9040	9042	9043	9044	9046	9047	9049	9051	9052
9054	9055	9057	9058	9060	9061	9062	9064	9065	9067	9069	9070	9072
9073	9075	9076	9079	9079	9081	9082	9084	9085	9087	9089	9092	9093
9095	9096	9098	9099	9101	9102	9107	9108	9110	9111	9113	9114	9116
9117	9119	9120	9122	9123	9125	9126	9128	9129	9132	9133	9135	9136
9138	9139	9141	9142	9145	9146	9148	9149	9151	9152	9154	9155	9157
9158	9160	9161	9163	9164	9166	9167	9169	9170	9172	9173	9178	9179
9181	9182	9184	9185	9187	9188	9190	9191	9193	9194	9196	9197	9199
9200	9203	9204	9206	9207	9209	9210	9212	9213	9216	9217	9219	9220
9223	9223	9224	9226	9228	9229	9231	9232	9234	9236	9237	9239	9240
9243	9244	9246	9247	9249	9250	9251	9253	9254	9256	9257	9259	9260
9263	9264	9266	9267	9269	9270	9271	9273	9274	9276	9277	9279	9280
9282	9283	9285	9286	9288	9289	9291	9292	9294	9296	9297	9299	9300
9303	9304	9306	9307	9309	9310	9311	9313	9314	9316	9317	9319	9320
9327	9329	9330	9332	9333	9335	9336	9338	9339	9341	9342	9344	9345
9347	9348	9349	9351	9352	9354	9355	9357	9358	9360	9361	9363	9365
9366	9367	9369	9371	9372	9374	9375	9377	9378	9380	9381	9383	9384
9386	9387	9389	9390	9392	9393	9395	9396	9398	9399	9401	9402	9405
9406	9409	9410	9412	9413	9415	9416	9418	9419	9421	9423	9427	9428
9430	9431	9433	9434	9436	9437	9439	9440	9442	9443	9445	9446	9449
9450	9452	9453	9455	9456	9457	9459	9460	9462	9463	9465	9466	9469
9471	9472	9474	9475	9477	9478	9480	9481	9483	9485	9486	9487	9489
9490	9495	9496	9499	9501	9502	9504	9505	9507	9509	9510	9512	9511
9513	9514	9516	9517	9520	9521	9523	9524	9526	9527	9529	9530	9533
9534	9536	9537	9539	9540	9541	9543	9544	9546	9547	9549	9550	9553
9554	9556	9557	9559	9560	9561	9563	9564	9566	9567	9569	9570	9599
9601	9602	9604	9605	9607	9608	9610	9611	9613	9614	9616	9617	9619
9620	9622	9623	9625	9626	9628	9629	9631	9632	9634	9635	9637	9638
9640	9641	9643	9644	9646	9648	9649	9650	9651	9653	9655	9656	9658
9659	9661	9662	9664	9665	9666	9668	9669	9670	9671	9673	9674	9677
9679	9680	9682	9683	9685	9686	9688	9689	9691	9692	9694	9695	9697
9698	9700	9701	9703	9704	9706	9707	9709	9710	9712	9713	9715	9716
9718	9719	9722	9723	9724	9726	9727	9729	9730	9732	9733	9736	9741
9742	9744	9745	9747	9748	9749	9750	9751	9753	9754	9756	9759	9760
9762	9763	9765	9767	9769	9770	9772	9772	9773	9775	9776	9780	9782
9783	9785	9786	9788	9789	9791	9792	9794	9794	9795	9797	9800	9801
9803	9804	9806	9807	9809	9812	9813	9815	9816	9818	9819	9822	9824
9825	9827	9828	9830	9831	9833	9833	9834	9837	9838	9840	9843	9844
9846	9847	9850	9851	9853	9853	9854	9856	9857	9859	9860	9862	9865
9867	9868	9870	9871	9873	9874	9877	9877	9878	9880	9881	9883	9885
9913	9915	9916	9918	9919	9921	9922	9924	9925	9927	9928	9930	9931
9933	9934	9936	9937	9939	9941	9942	9943	9945	9946	9948	9949	9951
9952	9954	9955	9957	9958	9960	9961	9963	9964	9966	9967	9969	9970
9972	9973	9975	9976	9978	9979	9981	9982	9984	9985	9987	9988	9990

CROSS REFERENCE TABLE -- USER SYMBOLS

9991	9993	9994	9995	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	10039	10040	10041	10044	10046	10047	10049	10050
10052	10053	10058	10059	10061	10062	10064	10065	10067	10068	10070	10071	10073
10074	10076	10077	10079	10080	10082	10084	10085	10087	10089	10090	10092	10093
10096	10097	10099	10100	10102	10103	10105	10106	10108	10109	10110	10111	10114
10115	10117	10118	10120	10121	10122	10123	10125	10126	10128	10129	10131	10132
10138	10139	10141	10142	10144	10145	10147	10148	10150	10151	10154	10155	10157
10158	10160	10161	10163	10164	10167	10168	10170	10171	10173	10174	10176	10177
10179	10180	10182	10184	10185	10187	10188	10191	10191	10194	10195	10197	10198
10200	10201	10229	10230	10232	10233	10235	10238	10239	10239	10241	10242	10244
10245	10247	10248	10250	10251	10253	10254	10257	10257	10259	10260	10262	10263
10265	10266	10268	10269	10271	10272	10274	10277	10277	10278	10280	10281	10283
10284	10286	10287	10289	10290	10292	10293	10293	10293	10294	10294	10291	10293
10304	10305	10307	10308	10310	10311	10313	10314	10315	10317	10319	10320	10322
10323	10325	10326	10328	10329	10331	10332	10334	10335	10337	10338	10340	10341
10343	10344	10345	10347	10349	10350	10352	10353	10355	10357	10360	10361	10363
10364	10366	10367	10369	10370	10375	10376	10377	10379	10381	10382	10384	10385
10387	10388	10390	10391	10393	10394	10395	10397	10400	10401	10402	10404	10406
10407	10409	10412	10413	10414	10416	10417	10418	10420	10422	10423	10425	10426
10428	10429	10432	10433	10434	10436	10437	10438	10440	10441	10442	10447	10449
10450	10452	10453	10454	10455	10458	10459	10461	10462	10464	10465	10467	10468
10471	10472	10473	10474	10477	10478	10480	10481	10484	10485	10487	10488	10490
10491	10492	10493	10494	10497	10499	10501	10502	10504	10505	10507	10508	10511
10512	10513	10514	10515	10518	10519	10521	10522	10524	10525	10527	10528	10531
10558	10559	10560	10561	10564	10565	10567	10568	10570	10571	10573	10574	10576
10577	10578	10579	10580	10583	10585	10586	10588	10589	10591	10592	10594	10595
10597	10598	10599	10600	10603	10604	10606	10607	10609	10610	10612	10613	10615
10616	10617	10618	10619	10621	10622	10624	10625	10627	10628	10630	10631	10634
10636	10637	10638	10640	10641	10643	10644	10645	10648	10649	10651	10652	10654
10655	10657	10658	10660	10661	10663	10664	10666	10667	10669	10670	10673	10674
10677	10678	10679	10680	10681	10684	10685	10687	10688	10693	10695	10696	10698
10699	10701	10702	10704	10705	10707	10708	10710	10711	10713	10714	10717	10718
10720	10721	10723	10724	10725	10727	10730	10731	10733	10734	10736	10737	10739
10740	10742	10743	10745	10746	10748	10749	10751	10752	10754	10755	10757	10758
10763	10764	10765	10766	10767	10769	10772	10773	10775	10776	10778	10779	10781
10782	10784	10785	10786	10789	10791	10792	10794	10795	10797	10798	10801	10802
10804	10805	10806	10808	10810	10811	10813	10814	10816	10818	10819	10821	10822
10824	10825	10826	10828	10831	10832	10834	10835	10836	10838	10839	10841	10842
10870	10872	10873	10874	10876	10878	10879	10881	10882	10884	10885	10887	10888
10890	10891	10892	10893	10895	10897	10898	10900	10902	10903	10905	10906	10908
10909	10911	10912	10913	10915	10917	10918	10920	10921	10923	10924	10926	10927
10929	10930	10932	10933	10935	10936	10938	10939	10941	10942	10944	10945	10947
10948	10950	10951	10953	10954	10956	10957	10959	10960	10962	10963	10965	10966
10968	10969	10971	10972	10974	10975	10977	10978	10980	10981	10983	10984	10986
10987	10990	10991	10994	10995	10997	10998	11000	11001	11003	11004	11009	11010
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	11180	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

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	13832	13833	13834	13836	13837	13839	13841	13847
13870	13871	13873	13874	13876	13877	13879	13880	13882	13883	13885	13886	13888
13889	13892	13893	13895	13896	13900	13901	13903	13904	13905	13907	13909	13910
13912	13913	13915	13916	13918	13919	13921	13922	13927	13928	13930	13931	13933
13934	13936	13937	13939	13940	13942	13943	13945	13946	13948	13949	13951	13952
13954	13955	13957	13958	13962	13963	13965	13966	13968	13969	13971	13972	13974
13975	13980	13981	13983	13984	13985	13987	13988	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	14031	14035	14037	14038
14040	14041	14046	14047	14049	14050	14053	14054	14056	14057	14059	14060	14063
14064	14065	14067	14069	14070	14073	14074	14075	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	14145	14146	14148	14149	14149	14151
14152	14155	14156	14158	14159	14161	14162	14165	14166	14168	14169	14171	14172
14175	14176	14178	14179	14181	14181	14185	14185	14189	14189	14191	14192	14195
14196	14198	14199	14204	14205	14207	14208	14211	14212	14214	14215	14217	14218
14221	14221	14223	14223	14225	14227	14230	14231	14233	14233	14235	14237	14239
14240	14243	14244	14246	14247	14249	14250	14252	14253	14255	14257	14259	14260
14262	14263	14265	14266	14269	14270	14272	14273	14275	14276	14278	14279	14282
14283	14285	14286	14288	14294	14295	14297	14298	14300	14300	14301	14301	14302
14303	14305	14306	14309	14311	14312	14314	14315	14317	14318	14321	14322	14324
14325	14327	14328	14330	14331	14334	14335	14337	14338	14343	14344	14346	14347
14349	14350	14352	14353	14356	14356	14361	14363	14364	14365	14367	14368	14370
14371	14373	14374	14376	14377	14380	14381	14383	14384	14385	14387	14389	14390
14392	14393	14395	14396	14398	14399	14401	14402	14404	14405	14408	14409	14411
14412	14414	14415	14417	14418	14420	14421	14425	14427	14429	14430	14432	14433
14436	14436	14438	14439	14441	14442	14444	14445	14448	14449	14451	14452	14455
14456	14458	14460	14461	14463	14465	14468	14470	14471	14473	14474	14476	14477
14479	14480	14482	14483	14485	14486	14488	14490	14491	14493	14494	14496	14497
14499	14500	14502	14503	14505	14506	14508	14510	14511	14512	14514	14515	14517
14519	14520	14521	14523	14524	14525	14527	14528	14530	14532	14533	14535	14536
14539	14540	14541	14543	14544	14545	14547	14548	14550	14551	14553	14554	14556
14557	14559	14560	14562	14563	14565	14566	14568	14569	14571	14572	14574	14575
14577	14578	14580	14581	14582	14584	14585	14587	14589	14590	14592	14593	14595
14596	14598	14599	14601	14602	14604	14605	14607	14608	14613	14614	14616	14617
14619	14620	14623	14624	14625	14627	14628	14633	14635	14636	14638	14639	14641
14642	14644	14645	14647	14648	14650	14651	14654	14655	14657	14658	14660	14661
14664	14665	14667	14668	14670	14671	14674	14670	14672	14673	14675	14676	14678
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	14838	14840	14843	14843	14845	14846	14848	14849	14851
14852	14854	14855	14857	14858	14860	14861	14863	14864	14866	14867	14869	14870
14872	14873	14875	14876	14878	14885	14889	14893	14894	14894	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	14950	14959	14961	14962
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

H12

DZKCA MACY11 27(1006) 13-MAY-77 14:07 PAGE 336

PAGE: 0357

DZKCA.P11 13-MAY-77 13:58

CROSS REFERENCE TABLE -- USER SYMBOLS

	15029	15031#	15032#	15034#	15035	15037#	15038#	15041#	15042	15044#	15045#	15047#	15048
.2A = 000120	15050#	15051#											
.2AWC = 000140	1#	1258#	12656										
	1#	5159	5229	11003	11071								

ROOMC	18	11636	11704																	
ROOM1	18																			
ROOM	18	3091	3180	3308	3406	7860	9734	9802	14534	14835	14905	14969	14990							
BB0	18	7409	7616	7643	7826	7987	8870	8921	14570	14853	15036									
BB1	18	14576	14591	14923																
BB2	18	13731	13759	13785	14558	14740														
BB7	18	2361	2441	2558	2638	2755	2835	2952	3032	3159	3248	3385	3483	3600	3680					
	18	3797	3877	3994	4074	4191	4271	4388	4468	4585	4665	4782	4862	4979	5059	5167				
	18	5237	5259	5444	5566	5651	5773	5858	5980	6065	6187	6272	6394	6479	6601	6686				
	18	6808	6893	7015	7100	7222	7307	8231	8388	13797	14546	14734	14760	14847	14856	14917				
	18	14926	14981	14996	15030															
BC	18																			
BZ	18																			
	18	3744	3824	3941	4021	4138	4218	4335	4415	4532	4612	4729	4809	4926	5006	5123				
	18	5193	5303	5388	5510	5595	5717	5802	5924	6009	6131	6216	6338	6423	6545	6630				
	18	6235	6352	6453	6575	6659	6781	6903	7025	7147	7269	7391	7513	7635	7757	7879				
	18	8235	8357	8479	8601	8723	8845	8967	9089	9211	9333	9455	9577	9699	9821	9943				
	18	10186	10377	10498	10619	10740	10861	10982	11103	11224	11345	11466	11587	11708	11829	11950				
	18	11771	11863	12033	12088	12279	12350	12431	12512	12667	12722	12913	12984	13039	13230	13301				
	18	13356	13547	13618	13673	14005	14363	14400	14437	14615	14634	14653	14672	14691	14710	14729				
CALL	18																			
	18	2128	2242	2312	2409	2488	2536	2584	2632	2680	2728	2776	2824	2872	2920	2968				
	18	3128	3242	3312	3409	3488	3536	3584	3632	3680	3728	3776	3824	3872	3920	3968				
	18	4128	4242	4312	4409	4488	4536	4584	4632	4680	4728	4776	4824	4872	4920	4968				
	18	5128	5242	5312	5409	5488	5536	5584	5632	5680	5728	5776	5824	5872	5920	5968				
	18	6128	6242	6312	6409	6488	6536	6584	6632	6680	6728	6776	6824	6872	6920	6968				
	18	7128	7242	7312	7409	7488	7536	7584	7632	7680	7728	7776	7824	7872	7920	7968				
	18	8128	8242	8312	8409	8488	8536	8584	8632	8680	8728	8776	8824	8872	8920	8968				
	18	9128	9242	9312	9409	9488	9536	9584	9632	9680	9728	9776	9824	9872	9920	9968				
	18	10128	10242	10312	10409	10488	10536	10584	10632	10680	10728	10776	10824	10872	10920	10968				
	18	11128	11242	11312	11409	11488	11536	11584	11632	11680	11728	11776	11824	11872	11920	11968				
	18	12128	12242	12312	12409	12488	12536	12584	12632	12680	12728	12776	12824	12872	12920	12968				
	18	13128	13242	13312	13409	13488	13536	13584	13632	13680	13728	13776	13824	13872	13920	13968				
	18	14128	14242	14312	14409	14488	14536	14584	14632	14680	14728	14776	14824	14872	14920	14968				
	18	15128	15242	15312	15409	15488	15536	15584	15632	15680	15728	15776	15824	15872	15920	15968				
	18	16128	16242	16312	16409	16488	16536	16584	16632	16680	16728	16776	16824	16872	16920	16968				
	18	17128	17242	17312	17409	17488	17536	17584	17632	17680	17728	17776	17824	17872	17920	17968				
	18	18128	18242	18312	18409	18488	18536	18584	18632	18680	18728	18776	18824	18872	18920	18968				
	18	19128	19242	19312	19409	19488	19536	19584	19632	19680	19728	19776	19824	19872	19920	19968				
	18	20128	20242	20312	20409	20488	20536	20584	20632	20680	20728	20776	20824	20872	20920	20968				
	18	21128	21242	21312	21409	21488	21536	21584	21632	21680	21728	21776	21824	21872	21920	21968				
	18	22128	22242	22312	22409	22488	22536	22584	22632	22680	22728	22776	22824	22872	22920	22968				
	18	23128	23242	23312	23409	23488	23536	23584	23632	23680	23728	23776	23824	23872	23920	23968				
	18	24128	24242	24312	24409	24488	24536	24584	24632	24680	24728	24776	24824	24872	24920	24968				
	18	25128	25242	25312	25409	25488	25536	25584	25632	25680	25728	25776	25824	25872	25920	25968				
	18	26128	26242	26312	26409	26488	26536	26584	26632	26680	26728	26776	26824	26872	26920	26968				
	18	27128	27242	27312	27409	27488	27536	27584	27632	27680	27728	27776	27824	27872	27920	27968				
	18	28128	28242	28312	28409	28488	28536	28584	28632	28680	28728	28776	28824	28872	28920	28968				
	18	29128	29242	29312	29409	29488	29536	29584	29632	29680	29728	29776	29824	29872	29920	29968				
	18	30128	30242	30312	30409	30488	30536	30584	30632	30680	30728	30776	30824	30872	30920	30968				
	18	31128	31242	31312	31409	31488	31536	31584	31632	31680	31728	31776	31824	31872	31920	31968				
	18	32128	32242	32312	32409	32488	32536	32584	32632	32680	32728	32776	32824	32872	32920	32968				
	18	33128	33242	33312	33409	33488	33536	33584	33632	33680	33728	33776	33824	33872	33920	33968				
	18	34128	34242	34312	34409	34488	34536	34584	34632	34680	34728	34776	34824	34872	34920	34968				
	18	35128	35242	35312	35409	35488	35536	35584	35632	35680	35728	35776	35824	35872	35920	35968				
	18	36128	36242	36312	36409	36488	36536	36584	36632	36680	36728	36776	36824	36872	36920	36968				
	18	37128	37242	37312	37409	37488	37536	37584	37632	37680	37728	37776	37824	37872	37920	37968				
	18	38128	38242	38312	38409	38488	38536	38584	38632	38680	38728	38776	38824	38872	38920	38968				
	18	39128	39242	39312	39409	39488	39536	39584	39632	39680	39728	39776	39824	39872	39920	39968				
	18	40128	40242	40312	40409	40488	40536	40584	40632	40680	40728	40776	40824	40872	40920	40968				
	18	41128	41242	41312	41409	41488	41536	41584	41632	41680	41728	41776	41824	41872	41920	41968				
	18	42128	42242	42312	42409	42488	42536	42584	42632	42680	42728	42776	42824	42872	42920	42968				
	18	43128	43242	43312	43409	43488	43536	43584	43632	43680	43728	43776	43824	43872	43920	43968				
	18	44128	44242	44312	44409	44488	44536	44584	44632	44680	44728	44776	44824	44872	44920	44968				
	18	45128	45242	45312	45409	45488	45536	45584	45632	45680	45728	45776	45824	45872	45920	45968				
	18	46128	46242	46312	46409	46488	46536	46584	46632	46680	46728	46776	46824	46872	46920	46968				
	18	47128	47242	47312	47409	47488	47536	47584	47632	47680	47728	47776	47824	47872	47920	47968				
	18	48128	48242	48312	48409	48488	48536	48584	48632	48680	48728	48776	48824	48872	48920	48968				
	18	49128	49242	49312	49409	49488	49536	49584	49632	49680	49728	49776	49824	49872	49920	49968				
	18	50128	50242	50312	50409	50488	50536	50584	50632	50680	50728	50776	50824	50872	50920	50968				
	18	51128	51242	51312	51409	51488	51536	51584	51632	51680	51728	51776	51824	51872	51920	51968				
	18	52128	52242	52312	52409	52488	52536	52584	52632	52680	52728	52776	5282							

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	2753	2767	2770	2773	2785	2788
2791	2794	2797	2800	2804	2810	2817	2823	2829	2834	2840	2876	2880	2884	2887
2890	2893	2895	2898	2901	2904	2911	2917	2927	2931	2940	2946	2950	2954	2957
2970	2973	2975	2978	2981	2984	2991	2997	3007	3011	3020	3026	3037	3071	3074
3078	3081	3083	3086	3089	3094	3099	3109	3115	3118	3121	3124	3128	3134	3141
3147	3150	3167	3171	3174	3177	3183	3189	3198	3201	3204	3207	3210	3213	3217
3223	3226	3236	3242	3245	3248	3251	3257	3267	3271	3274	3277	3281	3284	3287
3290	3293	3295	3298	3301	3304	3311	3317	3327	3331	3335	3338	3341	3344	3347
3350	3353	3363	3369	3372	3375	3381	3387	3397	3401	3404	3407	3410	3413	3417
3423	3426	3436	3442	3445	3448	3451	3457	3467	3471	3474	3477	3481	3484	3487
3490	3493	3495	3498	3501	3504	3511	3517	3527	3531	3535	3538	3541	3544	3547
3550	3553	3563	3569	3572	3575	3581	3587	3597	3601	3604	3607	3610	3613	3617
3623	3626	3636	3642	3645	3648	3651	3657	3667	3671	3674	3677	3681	3684	3687
3690	3693	3695	3698	3701	3704	3711	3717	3727	3731	3735	3738	3741	3744	3747
3750	3753	3763	3769	3772	3775	3781	3787	3797	3801	3804	3807	3810	3813	3817
3823	3826	3836	3842	3845	3848	3851	3857	3867	3871	3874	3877	3881	3884	3887
3890	3893	3895	3898	3901	3904	3911	3917	3927	3931	3935	3938	3941	3944	3947
3950	3953	3963	3969	3972	3975	3981	3987	3997	4001	4004	4007	4010	4013	4017
4023	4026	4036	4042	4045	4048	4051	4057	4067	4071	4074	4077	4081	4084	4087
4090	4093	4095	4098	4101	4104	4111	4117	4127	4131	4135	4138	4141	4144	4147
4150	4153	4163	4169	4172	4175	4181	4187	4197	4201	4204	4207	4210	4213	4217
4223	4226	4236	4242	4245	4248	4251	4257	4267	4271	4274	4277	4281	4284	4287
4290	4293	4303	4309	4312	4315	4321	4327	4337	4341	4345	4348	4351	4354	4357
4360	4363	4373	4379	4382	4385	4391	4397	4407	4411	4415	4418	4421	4424	4427
4430	4433	4443	4449	4452	4455	4461	4467	4477	4481	4485	4488	4491	4494	4497
4500	4503	4513	4519	4522	4525	4531	4537	4547	4551	4555	4558	4561	4564	4567
4570	4573	4583	4589	4592	4595	4601	4607	4617	4621	4625	4628	4631	4634	4637
4640	4643	4653	4659	4662	4665	4671	4677	4687	4691	4695	4698	4701	4704	4707
4710	4713	4723	4729	4732	4735	4741	4747	4757	4761	4765	4768	4771	4774	4777
4780	4783	4793	4799	4802	4805	4811	4817	4827	4831	4835	4838	4841	4844	4847
4850	4853	4863	4869	4872	4875	4881	4887	4897	4901	4905	4908	4911	4914	4917
4920	4923	4933	4939	4942	4945	4951	4957	4967	4971	4975	4978	4981	4984	4987
4990	4993	5003	5009	5012	5015	5021	5027	5037	5041	5045	5048	5051	5054	5057
5060	5063	5073	5079	5082	5085	5091	5097	5107	5111	5115	5118	5121	5124	5127
5130	5133	5143	5149	5152	5155	5161	5167	5177	5181	5185	5188	5191	5194	5197
5200	5203	5213	5219	5222	5225	5231	5237	5247	5251	5255	5258	5261	5264	5267
5270	5273	5283	5289	5292	5295	5301	5307	5317	5321	5325	5328	5331	5334	5337
5340	5343	5353	5359	5362	5365	5371	5377	5387	5391	5395	5398	5401	5404	5407
5410	5413	5423	5429	5432	5435	5441	5447	5457	5461	5465	5468	5471	5474	5477
5480	5483	5493	5499	5502	5505	5511	5517	5527	5531	5535	5538	5541	5544	5547
5550	5553	5563	5569	5572	5575	5581	5587	5597	5601	5605	5608	5611	5614	5617
5620	5623	5633	5639	5642	5645	5651	5657	5667	5671	5675	5678	5681	5684	5687
5690	5693	5703	5709	5712	5715	5721	5727	5737	5741	5745	5748	5751	5754	5757
5760	5763	5773	5779	5782	5785	5791	5797	5807	5811	5815	5818	5821	5824	5827
5830	5833	5843	5849	5852	5855	5861	5867	5877	5881	5885	5888	5891	5894	5897
5900	5903	5913	5919	5922	5925	5931	5937	5947	5951	5955	5958	5961	5964	5967
5970	5973	5983	5989	5992	5995	6001	6007	6017	6021	6025	6028	6031	6034	6037
6040	6043	6053	6059	6062	6065	6071	6077	6087	6091	6095	6098	6101	6104	6107
6110	6113	6123	6129	6132	6135	6141	6147	6157	6161	6165	6168	6171	6174	6177
6180	6183	6193	6199	6202	6205	6211	6217	6227	6231	6235	6238	6241	6244	6247
6250	6253	6263	6269	6272	6275	6281	6287	6297	6301	6305	6308	6311	6314	6317
6320	6323	6333	6339	6342	6345	6351	6357	6367	6371	6375	6378	6381	6384	6387
6390	6393	6403	6409	6412	6415	6421	6427	6437	6441	6445	6448	6451	6454	6457
6460	6463	6473	6479	6482	6485	6491	6497	6507	6511	6515	6518	6521	6524	6527
6530	6533	6543	6549	6552	6555	6561	6567	6577	6581	6585	6588	6591	6594	6597
6600	6603	6613	6619	6622	6625	6631	6637	6647	6651	6655	6658	6661	6664	6667
6670	6673	6683	6689	6692	6695	6701	6707	6717	6721	6725	6728	6731	6734	6737
6740	6743	6753	6759	6762	6765	6771	6777	6787	6791	6795	6798	6801	6804	6807
6810	6813	6823	6829	6832	6835	6841	6847	6857	6861	6865	6868	6871	6874	6877
6880	6883	6893	6899	6902	6905	6911	6917	6927	6931	6935	6938	6941	6944	6947
6950	6953	6963	6969	6972	6975	6981	6987	6997	7001	7005	7008	7011	7014	7017
7020	7023	7033	7039	7042	7045	7051	7057	7067	7071	7075	7078	7081	7084	7087
7090	7093	7103	7109	7112	7115	7121	7127	7137	7141	7145	7148	7151	7154	7157
7160	7163	7173	7179	7182	7185	7191	7197	7207	7211	7215	7218	7221	7224	7227
7230	7233	7243	7249	7252	7255	7261	7267	7277	7281	7285	7288	7291	7294	7297
7300	7303	7313	7319	7322	7325	7331	7337	7347	7351	7355	7358	7361	7364	7367
7370	7373	7383	7389	7392	7395	7401	7407	7417	7421	7425	7428	7431	7434	7437
7440	7443	7453	7459	7462	7465	7471	7477	7487	7491	7495	7498	7501	7504	7507
7510	7513	7523	7529	7532	7535	7541	7547	7557	7561	7565	7568	7571	7574	7577
7580	7583	7593	7599	7602	7605	7611	7617	7627	7631	7635	7638	7641	7644	7647
7650	7653	7663	7669	7672	7675	7681	7687	7697	7701	7705	7708	7711	7714	7717
7720	7723	7733	7739	7742	7745	7751	7757	7767	7771	7775	7778	7781	7784	7787
7790	7793	7803	7809	7812	7815	7821	7827	7837	7841	7845	7848	7851	7854	7857
7860	7863	7873	7879	7882	7885	7891	7897	7907	7911	7915	7918	7921	7924	7927
7930	7933	7943	7949	7952	7955	7961	7967	7977	7981	7985	7988	7991	7994	7997
8000	8003	8013	8019	8022	8025	8031	8037	8047	8051	8055	8058	8061	8064	8067
8070	8073	8083	8089	8092	8095	8101	8107	8117	8121	8125	8128	8131	8134	8137
8140	8143	8153	8159	8162	8165	8171	8177	8187	8191	8195	8198	8201	8204	8207
8210	8213	8223	8229	8232	8235	8241	8247	8257	8261	8265	8268	8271	8274	8277
8280	8283	8293	8299	8302	8305	8311	8317	8327	8331	8335	8338	8341	8344	8347
8350	8353	8363	8369	8372	8375	8381	8387	8397	8401	8405	8408	8411	8414	8417
8420	8423	8433	8439	8442	8445	8451	8457	8467	8471	8475	8478	8481	8484	8487
8490	8493	8503	8509	8512	8515	8521	8527	8537	8541	8545	8548	8551	8554	8557
8560	8563	8573	8579	8582	8585	8591	8597	8607	8611	8615	8618	8621	8624	8627
8630	8633	8643	8649	8652	8655	8661	8667	8677	8681	8685	8688	8691	8694	8697
8700	8703	8713	8719	8722	8725	8731	8737	8747	8751					

CROSS REFERENCE TABLE -- MACRO NAMES

11550	11553	11556	11559	11562	11565	11568	11571	11574	11577	11580	11583	11586	11589	11592
11595	11598	11601	11604	11607	11610	11613	11616	11619	11623	11627	11633	11640	11651	11654
11657	11660	11663	11667	11673	11680	11686	11692	11698	11698	11707	11719	11722	11725	11728
11731	11734	11738	11744	11751	11757	11763	11778	11781	11816	11819	11822	11825	11828	11831
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	11940	11944	11950	11955	11958	11971	11974	11977	11980	11984	11990
11997	12003	12009	12012	12015	12021	12026	12029	12032	12041	12048	12051	12055	12061	12069
12074	12080	12085	12130	12133	12136	12139	12142	12145	12148	12154	12157	12160	12163	12166
12169	12172	12175	12178	12181	12184	12187	12190	12193	12196	12199	12202	12205	12208	12211
12214	12217	12220	12223	12226	12229	12232	12235	12238	12241	12244	12247	12250	12253	12257
12261	12267	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	13607	13610	13613
13616	13619	13622	13625	13628	13631	13634	13637	13640	13643	13646	13649	13652	13655	13658
13661	13664	13667	13670	13673	13676	13679	13682	13685	13688	13691	13694	13697	13700	13703
13706	13709	13712	13715	13718	13721	13724	13727	13730	13733	13736	13739	13742	13745	13748
13751	13754	13757	13760	13763	13766	13769	13772	13775	13778	13781	13784	13787	13790	13793
13796	13799	13802	13805	13808	13811	13814	13817	13820	13823	13826	13829	13832	13835	13838
13841	13844	13847	13850	13853	13856	13859	13862	13865	13868	13871	13874	13877	13880	13883
13886	13889	13892	13895	13898	13901	13904	13907	13910	13913	13916	13919	13922	13925	13928
13931	13934	13937	13940	13943	13946	13949	13952	13955	13958	13961	13964	13967	13970	13973
13976	13979	13982	13985	13988	13991	13994	13997	14000	14003	14006	14009	14012	14015	14018
14021	14024	14027	14030	14033	14036	14039	14042	14045	14048	14051	14054	14057	14060	14063
14066	14069	14072	14075	14078	14081	14084	14087	14090	14093	14096	14099	14102	14105	14108
14111	14114	14117	14120	14123	14126	14129	14132	14135	14138	14141	14144	14147	14150	14153
14156	14159	14162	14165	14168	14171	14174	14177	14180	14183	14186	14189	14192	14195	14198
14201	14204	14207	14210	14213	14216	14219	14222	14225	14228	14231	14234	14237	14240	14243
14246	14249	14252	14255	14258	14261	14264	14267	14270	14273	14276	14279	14282	14285	14288
14291	14294	14297	14300	14303	14306	14309	14312	14315	14318	14321	14324	14327	14330	14333
14336	14339	14342	14345	14348	14351	14354	14357	14360	14363	14366	14369	14372	14375	14378
14381	14384	14387	14390	14393	14396	14399	14402	14405	14408	14411	14414	14417	14420	14423
14426	14429	14432	14435	14438	14441	14444	14447	14450	14453	14456	14459	14462	14465	14468
14471	14474	14477	14480	14483	14486	14489	14492	14495	14498	14501	14504	14507	14510	14513
14516	14519	14522	14525	14528	14531	14534	14537	14540	14543	14546	14549	14552	14555	14558
14561	14564	14567	14570	14573	14576	14579	14582	14585	14588	14591	14594	14597	14600	14603
14606	14609	14612	14615	14618	14621	14624	14627	14630	14633	14636	14639	14642	14645	14648
14651	14654	14657	14660	14663	14666	14669	14672	14675	14678	14681	14684	14687	14690	14693
14696	14699	14702	14705	14708	14711	14714	14717	14720	14723	14726	14729	14732	14735	14738
14741	14744	14747	14750	14753	14756	14759	14762	14765	14768	14771	14774	14777	14780	14783
14786	14789	14792	14795	14798	14801	14804	14807	14810	14813	14816	14819	14822	14825	14828
14831	14834	14837	14840	14843	14846	14849	14852	14855	14858	14861	14864	14867	14870	14873
14876	14879	14882	14885	14888	14891	14894	14897	14900	14903	14906	14909	14912	14915	14918
14921	14924	14927	14930	14933	14936	14939	14942	14945	14948	14951	14954	14957	14960	14963
14966	14969	14972	14975	14978	14981	14984	14987	14990	14993	14996	15000	15003	15006	15009
15012	15015	15018	15021	15024	15027	15030	15033	15036	15039	15042	15045	15048	15051	15054

MULT 176#
 NEXTST 176#
 OR 1#
 POP 14975
 PUSH 176#
 REPORT 1# 176#

3323 3421 7553 7610 7814 10051 10119 14540 14719 14728 14752 14826 14841 14911

DZKCA.P11 13-MAY-77 13:58 CROSS REFERENCE TABLE -- MACRO NAMES

SCOPE	718	15166													
SETPR	1768														
SETTRA	15448	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567	1568
	1569														
SETUP	1768														
SMBRT	18	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	1768														
SLASH	1768														
SPACE	1768														
STARS	1768	219	242	294	297	483	485	492	961	1028	1097	1176	1235	1241	1270
	1338	1523	1664	1680											
SUBMC	18	11953	12021												
SUB1	18														
SUB1C	18	8541	8625	8701	8791	11319	11387								
SUB2C	18	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	1768														
TRNTRP	15448														
TYP9IN	1768														
TYPDEC	1768														
TYPNAM	1768														
TY-NUM	1768														
TYPOCS	1768														
TYPOCT	1768														
TYPTXT	1768														
\$ADC	18	5161	5231	8008	8096	8237	8397	8532	8616	8695	8782	8864	8915	12904	12972
	13791	13967	14769	14772											
\$ADD	18	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	13285	13351
	13532	13603	13668	14862	14932										
\$ALUTO	308	8811													
\$ALUTI	308	8934	9251	9568	9885	10202	10519	10836	11153	11470	11787	12104	12421	12738	13055
	13372														
\$ASL	18	12587	12655												
\$AUTO	18	817													
\$BEGIN	18	2281	2478	2675	2872	3070	3267	3520	3717	3914	4111	4308	4505	4702	4899
	5097	5274	5481	5688	5895	6102	6309	6516	6723	6930	7137	7343	7517	7760	7997
	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					
\$BR	18	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	2854	2859	2874	2877	3004	3004	3010	3017	3023	3040	3043
	3131	3137	3144	3150	3162	3220	3226	3232	3256	3259	3259	3357	3363	3370	3376
	3388	3455	3461	3468	3474	3491	3494	3572	3585	3591	3603	3652	3658	3665	3665
	3671	3688	3691	3769	3775	3782	3788	3794	3855	3855	3862	3868	3885	3888	3966
	3972	3979	3985	3997	4046	4052	4059	4065	4085	4163	4169	4176	4182	4188	4194
	4243	4249	4256	4262	4279	4282	4360			4379	4391	4440	4446	4453	4459

CROSS REFERENCE TABLE -- MACRO NAMES

	4476	4479	4527	4532	4570	4575	4588	4627	4643	4650	4655	4673	4676	4754	4760
	4767	4773	4785	4832	4840	4847	4858	4870	4873	4875	4876	4877	4878	4954	5031
	5013	5044	5050	5067	5070	5139	5144	5149	5151	5152	5153	5154	5155	5225	5245
	5257	5261	5262	5263	5264	5265	5266	5267	5268	5269	5270	5271	5272	5321	5322
	5273	5274	5275	5276	5277	5278	5279	5280	5281	5282	5283	5284	5285	5323	5324
	5286	5287	5288	5289	5290	5291	5292	5293	5294	5295	5296	5297	5298	5325	5326
	5299	5300	5301	5302	5303	5304	5305	5306	5307	5308	5309	5310	5311	5327	5328
	5312	5313	5314	5315	5316	5317	5318	5319	5320	5321	5322	5323	5324	5329	5330
	5331	5332	5333	5334	5335	5336	5337	5338	5339	5340	5341	5342	5343	5331	5332
	5344	5345	5346	5347	5348	5349	5350	5351	5352	5353	5354	5355	5356	5333	5334
	5357	5358	5359	5360	5361	5362	5363	5364	5365	5366	5367	5368	5369	5335	5336
	5370	5371	5372	5373	5374	5375	5376	5377	5378	5379	5380	5381	5382	5337	5338
	5383	5384	5385	5386	5387	5388	5389	5390	5391	5392	5393	5394	5395	5339	5340
	5396	5397	5398	5399	5400	5401	5402	5403	5404	5405	5406	5407	5408	5341	5342
	5409	5410	5411	5412	5413	5414	5415	5416	5417	5418	5419	5420	5421	5343	5344
	5422	5423	5424	5425	5426	5427	5428	5429	5430	5431	5432	5433	5434	5345	5346
	5435	5436	5437	5438	5439	5440	5441	5442	5443	5444	5445	5446	5447	5347	5348
	5448	5449	5450	5451	5452	5453	5454	5455	5456	5457	5458	5459	5460	5349	5350
	5461	5462	5463	5464	5465	5466	5467	5468	5469	5470	5471	5472	5473	5351	5352
	5474	5475	5476	5477	5478	5479	5480	5481	5482	5483	5484	5485	5486	5353	5354
	5487	5488	5489	5490	5491	5492	5493	5494	5495	5496	5497	5498	5499	5355	5356
	5500	5501	5502	5503	5504	5505	5506	5507	5508	5509	5510	5511	5512	5357	5358
	5513	5514	5515	5516	5517	5518	5519	5520	5521	5522	5523	5524	5525	5359	5360
	5526	5527	5528	5529	5530	5531	5532	5533	5534	5535	5536	5537	5538	5361	5362
	5539	5540	5541	5542	5543	5544	5545	5546	5547	5548	5549	5550	5551	5363	5364
	5552	5553	5554	5555	5556	5557	5558	5559	5560	5561	5562	5563	5564	5365	5366
	5565	5566	5567	5568	5569	5570	5571	5572	5573	5574	5575	5576	5577	5367	5368
	5578	5579	5580	5581	5582	5583	5584	5585	5586	5587	5588	5589	5590	5369	5370
	5591	5592	5593	5594	5595	5596	5597	5598	5599	5600	5601	5602	5603	5371	5372
	5604	5605	5606	5607	5608	5609	5610	5611	5612	5613	5614	5615	5616	5373	5374
	5617	5618	5619	5620	5621	5622	5623	5624	5625	5626	5627	5628	5629	5375	5376
	5630	5631	5632	5633	5634	5635	5636	5637	5638	5639	5640	5641	5642	5377	5378
	5643	5644	5645	5646	5647	5648	5649	5650	5651	5652	5653	5654	5655	5379	5380
	5656	5657	5658	5659	5660	5661	5662	5663	5664	5665	5666	5667	5668	5381	5382
	5669	5670	5671	5672	5673	5674	5675	5676	5677	5678	5679	5680	5681	5383	5384
	5682	5683	5684	5685	5686	5687	5688	5689	5690	5691	5692	5693	5694	5385	5386
	5695	5696	5697	5698	5699	5700	5701	5702	5703	5704	5705	5706	5707	5387	5388
	5708	5709	5710	5711	5712	5713	5714	5715	5716	5717	5718	5719	5720	5389	5390
	5721	5722	5723	5724	5725	5726	5727	5728	5729	5730	5731	5732	5733	5391	5392
	5734	5735	5736	5737	5738	5739	5740	5741	5742	5743	5744	5745	5746	5393	5394
	5747	5748	5749	5750	5751	5752	5753	5754	5755	5756	5757	5758	5759	5395	5396
	5760	5761	5762	5763	5764	5765	5766	5767	5768	5769	5770	5771	5772	5397	5398
	5773	5774	5775	5776	5777	5778	5779	5780	5781	5782	5783	5784	5785	5399	5400
	5786	5787	5788	5789	5790	5791	5792	5793	5794	5795	5796	5797	5798	5401	5402
	5799	5800	5801	5802	5803	5804	5805	5806	5807	5808	5809	5810	5811	5403	5404
	5812	5813	5814	5815	5816	5817	5818	5819	5820	5821	5822	5823	5824	5405	5406
	5825	5826	5827	5828	5829	5830	5831	5832	5833	5834	5835	5836	5837	5407	5408
	5838	5839	5840	5841	5842	5843	5844	5845	5846	5847	5848	5849	5850	5409	5410
	5851	5852	5853	5854	5855	5856	5857	5858	5859	5860	5861	5862	5863	5411	5412
	5864	5865	5866	5867	5868	5869	5870	5871	5872	5873	5874	5875	5876	5413	5414
	5877	5878	5879	5880	5881	5882	5883	5884	5885	5886	5887	5888	5889	5415	5416
	5890	5891	5892	5893	5894	5895	5896	5897	5898	5899	5900	5901	5902	5417	5418
	5903	5904	5905	5906	5907	5908	5909	5910	5911	5912	5913	5914	5915	5419	5420
	5916	5917	5918	5919	5920	5921	5922	5923	5924	5925	5926	5927	5928	5421	5422
	5929	5930	5931	5932	5933	5934	5935	5936	5937	5938	5939	5940	5941	5423	5424
	5942	5943	5944	5945	5946	5947	5948	5949	5950	5951	5952	5953	5954	5425	5426
	5955	5956	5957	5958	5959	5960	5961	5962	5963	5964	5965	5966	5967	5427	5428
	5968	5969	5970	5971	5972	5973	5974	5975	5976	5977	5978	5979	5980	5429	5430
	5981	5982	5983	5984	5985	5986	5987	5988	5989	5990	5991	5992	5993	5431	5432
	5994	5995	5996	5997	5998	5999	6000	6001	6002	6003	6004	6005	6006	5433	5434
	6007	6008	6009	6010	6011	6012	6013	6014	6015	6016	6017	6018	6019	5435	5436
	6020	6021	6022	6023	6024	6025	6026	6027	6028	6029	6030	6031	6032	5437	5438
	6033	6034	6035	6036	6037	6038	6039	6040	6041	6042	6043	6044	6045	5439	5440
	6046	6047	6048	6049	6050	6051	6052	6053	6054	6055	6056	6057	6058	5441	5442
	6059	6060	6061	6062	6063	6064	6065	6066	6067	6068	6069	6070	6071	5443	5444
	6072	6073	6074	6075	6076	6077	6078	6079	6080	6081	6082	6083	6084	5445	5446
	6085	6086	6087	6088	6089	6090	6091	6092	6093	6094	6095	6096	6097	5447	5448
	6098	6099	6100	6101	6102	6103	6104	6105	6106	6107	6108	6109	6110	5449	5450
	6111	6112	6113	6114	6115	6116	6117	6118	6119	6120	6121	6122	6123	5451	5452
	6124	6125	6126	6127	6128	6129	6130	6131	6132	6133	6134	6135	6136	5453	5454
	6137	6138	6139	6140	6141	6142	6143	6144	6145	6146	6147	6148	6149	5455	5456
	6150	6151	6152	6153	6154	6155	6156	6157	6158	6159	6160	6161	6162	5457	5458
	6163	6164	6165	6166	6167	6168	6169	6170	6171	6172	6173	6174	6175	5459	5460
	6176	6177	6178	6179	6180	6181	6182	6183	6184	6185	6186	6187	6188	5461	5462
	6189	6190	6191	6192	6193	6194	6195	6196	6197	6198	6199	6200	6201	5463	5464
	6202	6203	6204	6205	6206	6207	6208	6209	6210	6211	6212	6213	6214	5465	5466
	6215	6216	6217	6218	6219	6220	6221	6222	6223	6224	6225	6226	6227	5467	5468
	6228	6229	6230	6231	6232	6233	6234	6235	6236	6237	6238	6239	6240	5469	5470
	6241	6242	6243	6244	6245	6246	6247	6248	6249	6250	6251	6252	6253	5471	5472
	6254	6255	6256	6257	6258	6259	6260	6261	6262	6263	6264	6265	6266	5473	5474
	6267	6268	6269	6270	6271	6272	6273	6274	6275	6276	6277	6278	6279	5475	5476
	6280	6281	6282	6283	6284	6285	6286	6287	6288	6289	6290	6291	6292	5477	5478
	6293	6294	6295	6296	6297	6298	6299	6300	6301	6302	6303	6304	6305	5479	5480
	6306	6307	6308	6309	6310	6311	6312	6313	6314	6315	6316	6317	6318	5481	5482
	6319	6320	6321												

CROSS REFERENCE TABLE -- MACRO NAMES

\$IFLOS	1#													
\$IFNE	1#	14609												
\$INC	1#	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	1#													
\$LOC	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	14807				
\$CLICK	29#													
\$PARMI	1#													
\$MEM10	21#	8115												
\$MEM11	19#	8272												
\$MEM2	22#	8432												
\$MEM31	23#	8645												
\$MFLT	18#	8137	8294											
\$MOCK	1#													
\$MSG	1#	1769												
\$PAUSE	30#	13830												
\$PAC	28#													
\$PART	25#	7321												
\$PSE	26#	7494												
\$PQ	27#	7873												
\$P	8#	7737												
\$	1#	967												
\$	1#	1659												
\$	29#	13689												
\$	6#													
\$	9#	5073												
\$MFL	30#	14673												
\$QUEST	1#	1967	1979	1987	2044	2052								
\$RANCL	1#	1713												
\$RCLK	1#	1716	1719	1756	1761									
\$ROL	1#	5158	5228	11002	11070									
\$ROVAR	1#	363												
\$SCAD0	1#	1039												
\$SCAD1	1#	1078												
\$SIMBC	1#													
\$SOFTC	1#	1789												
\$SPFLT	12#	5281	5366	5488	5573	5695	5780	5902	5987	6109	6194	6316	6401	6523 6608
	6730	6815	6937	7022	7144	7229								
\$SPTS1	13#	5251	5458	5665	5872	6079	6286	6493	6700	6907	7114			
\$SP2	14#													
\$SUBR	1#													
\$STSN	1#													
	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					
\$UPAD0	1#	1687												
\$VARIA	1#	235												
\$XOR	1#	10368	10436											
\$XZ	1#	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

DZKCA MACY11 27(1006) 13-MAY-77 14:07 PAGE 345
DZKCA.P11 13-MAY-77 13:58 CROSS REFERENCE TABLE -- MACRO NAMES

SSCURE
SSCUTM
SSSCA
SSSEMT
SSSCOP
SSST
SSXIP
SSPAGE
SSRCTL

12115	12432	10203	10213	10520	10530	10837	10847	11154	11164	11471	11481	11788	11798	12105
12432	12739	12432	12739	12749	13056	13066	13373	13383	13690	13698	13831	13841	14458	14465
280	281	280	281	282	283	284								
1556	1557	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567	1568
10101	10102	10103	10104	10105	10106	10107	10108	10109	10110	10111	10112	10113	10114	10115
10116	10117	10118	10119	10120	10121	10122	10123	10124	10125	10126	10127	10128	10129	10130
10131	10132	10133	10134	10135	10136	10137	10138	10139	10140	10141	10142	10143	10144	10145
10146	10147	10148	10149	10150	10151	10152	10153	10154	10155	10156	10157	10158	10159	10160
10161	10162	10163	10164	10165	10166	10167	10168	10169	10170	10171	10172	10173	10174	10175
10176	10177	10178	10179	10180	10181	10182	10183	10184	10185	10186	10187	10188	10189	10190
10191	10192	10193	10194	10195	10196	10197	10198	10199	10200	10201	10202	10203	10204	10205
10206	10207	10208	10209	10210	10211	10212	10213	10214	10215	10216	10217	10218	10219	10220
10221	10222	10223	10224	10225	10226	10227	10228	10229	10230	10231	10232	10233	10234	10235
10236	10237	10238	10239	10240	10241	10242	10243	10244	10245	10246	10247	10248	10249	10250
10251	10252	10253	10254	10255	10256	10257	10258	10259	10260	10261	10262	10263	10264	10265
10266	10267	10268	10269	10270	10271	10272	10273	10274	10275	10276	10277	10278	10279	10280
10281	10282	10283	10284	10285	10286	10287	10288	10289	10290	10291	10292	10293	10294	10295
10296	10297	10298	10299	10300	10301	10302	10303	10304	10305	10306	10307	10308	10309	10310
10311	10312	10313	10314	10315	10316	10317	10318	10319	10320	10321	10322	10323	10324	10325
10326	10327	10328	10329	10330	10331	10332	10333	10334	10335	10336	10337	10338	10339	10340
10341	10342	10343	10344	10345	10346	10347	10348	10349	10350	10351	10352	10353	10354	10355
10356	10357	10358	10359	10360	10361	10362	10363	10364	10365	10366	10367	10368	10369	10370
10371	10372	10373	10374	10375	10376	10377	10378	10379	10380	10381	10382	10383	10384	10385
10386	10387	10388	10389	10390	10391	10392	10393	10394	10395	10396	10397	10398	10399	10400
10401	10402	10403	10404	10405	10406	10407	10408	10409	10410	10411	10412	10413	10414	10415
10416	10417	10418	10419	10420	10421	10422	10423	10424	10425	10426	10427	10428	10429	10430
10431	10432	10433	10434	10435	10436	10437	10438	10439	10440	10441	10442	10443	10444	10445
10446	10447	10448	10449	10450	10451	10452	10453	10454	10455	10456	10457	10458	10459	10460
10461	10462	10463	10464	10465	10466	10467	10468	10469	10470	10471	10472	10473	10474	10475
10476	10477	10478	10479	10480	10481	10482	10483	10484	10485	10486	10487	10488	10489	10490
10491	10492	10493	10494	10495	10496	10497	10498	10499	10500	10501	10502	10503	10504	10505
10506	10507	10508	10509	10510	10511	10512	10513	10514	10515	10516	10517	10518	10519	10520
10521	10522	10523	10524	10525	10526	10527	10528	10529	10530	10531	10532	10533	10534	10535
10536	10537	10538	10539	10540	10541	10542	10543	10544	10545	10546	10547	10548	10549	10550
10551	10552	10553	10554	10555	10556	10557	10558	10559	10560	10561	10562	10563	10564	10565
10566	10567	10568	10569	10570	10571	10572	10573	10574	10575	10576	10577	10578	10579	10580
10581	10582	10583	10584	10585	10586	10587	10588	10589	10590	10591	10592	10593	10594	10595
10596	10597	10598	10599	10600	10601	10602	10603	10604	10605	10606	10607	10608	10609	10610
10611	10612	10613	10614	10615	10616	10617	10618	10619	10620	10621	10622	10623	10624	10625
10626	10627	10628	10629	10630	10631	10632	10633	10634	10635	10636	10637	10638	10639	10640
10641	10642	10643	10644	10645	10646	10647	10648	10649	10650	10651	10652	10653	10654	10655
10656	10657	10658	10659	10660	10661	10662	10663	10664	10665	10666	10667	10668	10669	10670
10671	10672	10673	10674	10675	10676	10677	10678	10679	10680	10681	10682	10683	10684	10685
10686	10687	10688	10689	10690	10691	10692	10693	10694	10695	10696	10697	10698	10699	10700
10701	10702	10703	10704	10705	10706	10707	10708	10709	10710	10711	10712	10713	10714	10715
10716	10717	10718	10719	10720	10721	10722	10723	10724	10725	10726	10727	10728	10729	10730
10731	10732	10733	10734	10735	10736	10737	10738	10739	10740	10741	10742	10743	10744	10745
10746	10747	10748	10749	10750	10751	10752	10753	10754	10755	10756	10757	10758	10759	10760
10761	10762	10763	10764	10765	10766	10767	10768	10769	10770	10771	10772	10773	10774	10775
10776	10777	10778	10779	10780	10781	10782	10783	10784	10785	10786	10787	10788	10789	10790
10791	10792	10793	10794	10795	10796	10797	10798	10799	10800	10801	10802	10803	10804	10805
10806	10807	10808	10809	10810	10811	10812	10813	10814	10815	10816	10817	10818	10819	10820
10821	10822	10823	10824	10825	10826	10827	10828	10829	10830	10831	10832	10833	10834	10835
10836	10837	10838	10839	10840	10841	10842	10843	10844	10845	10846	10847	10848	10849	10850
10851	10852	10853	10854	10855	10856	10857	10858	10859	10860	10861	10862	10863	10864	10865
10866	10867	10868	10869	10870	10871	10872	10873	10874	10875	10876	10877	10878	10879	10880
10881	10882	10883	10884	10885	10886	10887	10888	10889	10890	10891	10892	10893	10894	10895
10896	10897	10898	10899	10900	10901	10902	10903	10904	10905	10906	10907	10908	10909	10910
10911	10912	10913	10914	10915	10916	10917	10918	10919	10920	10921	10922	10923	10924	10925
10926	10927	10928	10929	10930	10931	10932	10933	10934	10935	10936	10937	10938	10939	10940
10941	10942	10943	10944	10945	10946	10947	10948	10949	10950	10951	10952	10953	10954	10955
10956	10957	10958	10959	10960	10961	10962	10963	10964	10965	10966	10967	10968	10969	10970
10971	10972	10973	10974	10975	10976	10977	10978	10979	10980	10981	10982	10983	10984	10985
10986	10987	10988	10989	10990	10991	10992	10993	10994	10995	10996	10997	10998	10999	11000
11001	11002	11003	11004	11005	11006	11007	11008	11009	11010	11011	11012	11013	11014	11015
11016	11017	11018	11019	11020	11021	11022	11023	11024	11025	11026	11027	11028	11029	11030
11031	11032	11033	11034	11035	11036	11037	11038	11039	11040	11041	11042	11043	11044	11045
11046	11047	11048	11049	11050	11051	11052	11053	11054	11055	11056	11057	11058	11059	11060
11061	11062	11063	11064	11065	11066	11067	11068	11069	11070	11071	11072	11073	11074	11075
11076	11077	11078	11079	11080	11081	11082	11083	11084	11085	11086	11087	11088	11089	11090
11091	11092	11093	11094	11095	11096	11097	11098	11099	11100	11101	11102			

CROSS REFERENCE TABLE -- MACRO NAMES

13340	13619
13663	13780
13815	14025
14076	14188
14276	14388
14411	14501
14557	14616
14718	14857
2337	
13346	13644
13367	13786
13387	14048
13397	14189
13407	14389
13417	14501
13427	14616
13437	14857
13447	
13457	
13467	
13477	
13487	
13497	
13507	
13517	
13527	
13537	
13547	
13557	
13567	
13577	
13587	
13597	
13607	
13617	
13627	
13637	
13647	
13657	
13667	
13677	
13687	
13697	
13707	
13717	
13727	
13737	
13747	
13757	
13767	
13777	
13787	
13797	
13807	
13817	
13827	
13837	
13847	
13857	
13867	
13877	
13887	
13897	
13907	
13917	
13927	
13937	
13947	
13957	
13967	
13977	
13987	
13997	
14007	
14017	
14027	
14037	
14047	
14057	
14067	
14077	
14087	
14097	
14107	
14117	
14127	
14137	
14147	
14157	
14167	
14177	
14187	
14197	
14207	
14217	
14227	
14237	
14247	
14257	
14267	
14277	
14287	
14297	
14307	
14317	
14327	
14337	
14347	
14357	
14367	
14377	
14387	
14397	
14407	
14417	
14427	
14437	
14447	
14457	
14467	
14477	
14487	
14497	
14507	
14517	
14527	
14537	
14547	
14557	
14567	
14577	
14587	
14597	
14607	
14617	
14627	
14637	
14647	
14657	
14667	
14677	
14687	
14697	
14707	
14717	
14727	
14737	
14747	
14757	
14767	
14777	
14787	
14797	
14807	
14817	
14827	
14837	
14847	
14857	
14867	
14877	
14887	
14897	
14907	
14917	
14927	
14937	
14947	
14957	
14967	
14977	
14987	
14997	
15007	
15017	
15027	
15037	
15047	
15057	
15067	
15077	
15087	
15097	
15107	
15117	
15127	
15137	
15147	
15157	
15167	
15177	
15187	
15197	
15207	
15217	
15227	
15237	
15247	
15257	
15267	
15277	
15287	
15297	
15307	
15317	
15327	
15337	
15347	
15357	
15367	
15377	
15387	
15397	
15407	
15417	
15427	
15437	
15447	
15457	
15467	
15477	
15487	
15497	
15507	
15517	
15527	
15537	
15547	
15557	
15567	
15577	
15587	
15597	
15607	
15617	
15627	
15637	
15647	
15657	
15667	
15677	
15687	
15697	
15707	
15717	
15727	
15737	
15747	
15757	
15767	
15777	
15787	
15797	
15807	
15817	
15827	
15837	
15847	
15857	
15867	
15877	
15887	
15897	
15907	
15917	
15927	
15937	
15947	
15957	
15967	
15977	
15987	
15997	
16007	
16017	
16027	
16037	
16047	
16057	
16067	
16077	
16087	
16097	
16107	
16117	
16127	
16137	
16147	
16157	
16167	
16177	
16187	
16197	
16207	
16217	
16227	
16237	
16247	
16257	
16267	
16277	
16287	
16297	
16307	
16317	
16327	
16337	
16347	
16357	
16367	
16377	
16387	
16397	
16407	
16417	
16427	
16437	
16447	
16457	
16467	
16477	
16487	
16497	
16507	
16517	
16527	
16537	
16547	
16557	
16567	
16577	
16587	
16597	
16607	
16617	
16627	
16637	
16647	
16657	
16667	
16677	
16687	
16697	
16707	
16717	
16727	
16737	
16747	
16757	
16767	
16777	
16787	
16797	
16807	
16817	
16827	
16837	
16847	
16857	
16867	
16877	
16887	
16897	
16907	
16917	
16927	
16937	
16947	
16957	
16967	
16977	
16987	
16997	
17007	
17017	
17027	
17037	
17047	
17057	
17067	
17077	
17087	
17097	
17107	
17117	
17127	
17137	
17147	
17157	
17167	
17177	
17187	
17197	
17207	
17217	
17227	
17237	
17247	
17257	
17267	
17277	
17287	
17297	
17307	
17317	
17327	
17337	
17347	
17357	
17367	
17377	
17387	
17397	
17407	
17417	
17427	
17437	
17447	
17457	
17467	
17477	
17487	
17497	
17507	
17517	
17527	
17537	
17547	
17557	
17567	
17577	
17587	
17597	
17607	
17617	
17627	
17637	
17647	
17657	
17667	
17677	
17687	
17697	
17707	
17717	
17727	
17737	
17747	
17757	
17767	
17777	
17787	
17797	
17807	
17817	
17827	
17837	
17847	
17857	
17867	
17877	
17887	
17897	
17907	
17917	
17927	
17937	
17947	
17957	
17967	
17977	
17987	
17997	
18007	
18017	
18027	
18037	
18047	
18057	
18067	
18077	
18087	
18097	
18107	
18117	
18127	
18137	
18147	
18157	
18167	
18177	
18187	
18197	
18207	
18217	
18227	
18237	
18247	
18257	
18267	
18277	
18287	
18297	
18307	
18317	
18327	
18337	
18347	
18357	
18367	
18377	
18387	
18397	
18407	
18417	
18427	
18437	
18447	
18457	
18467	
18477	
18487	
18497	
18507	
18517	
18527	
18537	
18547	
18557	
18567	
18577	
18587	
18597	
18607	
18617	
18627	
18637	
18647	
18657	
18667	
18677	
18687	
18697	
18707	
18717	
18727	
18737	
18747	
18757	
18767	
18777	
18787	
18797	
18807	
18817	
18827	
18837	
18847	
18857	
18867	
18877	
18887	
18897	
18907	
18917	
18927	
18937	
18947	
18957	
18967	
18977	
18987	
18997	
19007	
19017	
19027	
19037	
19047	
19057	
19067	
19077	
19087	
19097	
19107	
19117	
19127	
19137	
19147	
19157	
19167	
19177	
19187	
19197	
19207	
19217	
19227	
19237	
19247	
19257	
19267	
19277	
19287	
19297	
19307	
19317	
19327	
19337	
19347	
19357	
19367	
19377	
19387	
19397	
19407	
19417	
19427	
19437	
19447	
19457	
19467	
19477	
19487	
19497	
19507	
19517	
19527	
19537	
19547	
19557	
19567	
19577	
19587	
19597	
19607	

CROSS REFERENCE TABLE -- MACRO NAMES

10313	10316	10319	10322	10325	10328	10331	10334	10337	10340	10343	10346	10349	10352	10356
10360	10366	10369	10372	10375	10378	10381	10384	10387	10390	10393	10396	10399	10402	10405
10412	10418	10421	10424	10427	10430	10433	10436	10439	10442	10445	10448	10451	10454	10457
10460	10466	10469	10472	10475	10478	10481	10484	10487	10490	10493	10496	10499	10502	10505
10512	10518	10521	10524	10527	10530	10533	10536	10539	10542	10545	10548	10551	10554	10557
10560	10566	10569	10572	10575	10578	10581	10584	10587	10590	10593	10596	10599	10602	10605
10612	10618	10621	10624	10627	10630	10633	10636	10639	10642	10645	10648	10651	10654	10657
10660	10666	10669	10672	10675	10678	10681	10684	10687	10690	10693	10696	10699	10702	10705
10712	10718	10721	10724	10727	10730	10733	10736	10739	10742	10745	10748	10751	10754	10757
10760	10766	10769	10772	10775	10778	10781	10784	10787	10790	10793	10796	10799	10802	10805
10812	10818	10821	10824	10827	10830	10833	10836	10839	10842	10845	10848	10851	10854	10857
10860	10866	10869	10872	10875	10878	10881	10884	10887	10890	10893	10896	10899	10902	10905
10912	10918	10921	10924	10927	10930	10933	10936	10939	10942	10945	10948	10951	10954	10957
10960	10966	10969	10972	10975	10978	10981	10984	10987	10990	10993	10996	10999	11002	11005
10999	11005	11008	11011	11014	11017	11020	11023	11026	11029	11032	11035	11038	11041	11044
11047	11053	11056	11059	11062	11065	11068	11071	11074	11077	11080	11083	11086	11089	11092
11095	11101	11104	11107	11110	11113	11116	11119	11122	11125	11128	11131	11134	11137	11140
11143	11146	11149	11152	11155	11158	11161	11164	11167	11170	11173	11176	11179	11182	11185
11188	11191	11194	11197	11200	11203	11206	11209	11212	11215	11218	11221	11224	11227	11230
11233	11236	11239	11242	11245	11248	11251	11254	11257	11260	11263	11266	11269	11272	11275
11278	11281	11284	11287	11290	11293	11296	11299	11302	11305	11308	11311	11314	11317	11320
11323	11326	11329	11332	11335	11338	11341	11344	11347	11350	11353	11356	11359	11362	11365
11368	11371	11374	11377	11380	11383	11386	11389	11392	11395	11398	11401	11404	11407	11410
11413	11416	11419	11422	11425	11428	11431	11434	11437	11440	11443	11446	11449	11452	11455
11458	11461	11464	11467	11470	11473	11476	11479	11482	11485	11488	11491	11494	11497	11500
11503	11506	11509	11512	11515	11518	11521	11524	11527	11530	11533	11536	11539	11542	11545
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
11638	11641	11644	11647	11650	11653	11656	11659	11662	11665	11668	11671	11674	11677	11680
11683	11686	11689	11692	11695	11698	11701	11704	11707	11710	11713	11716	11719	11722	11725
11728	11731	11734	11737	11740	11743	11746	11749	11752	11755	11758	11761	11764	11767	11770
11773	11776	11779	11782	11785	11788	11791	11794	11797	11800	11803	11806	11809	11812	11815
11818	11821	11824	11827	11830	11833	11836	11839	11842	11845	11848	11851	11854	11857	11860
11863	11866	11869	11872	11875	11878	11881	11884	11887	11890	11893	11896	11899	11902	11905
11908	11911	11914	11917	11920	11923	11926	11929	11932	11935	11938	11941	11944	11947	11950
11953	11956	11959	11962	11965	11968	11971	11974	11977	11980	11983	11986	11989	11992	11995
11998	12001	12004	12007	12010	12013	12016	12019	12022	12025	12028	12031	12034	12037	12040
12043	12046	12049	12052	12055	12058	12061	12064	12067	12070	12073	12076	12079	12082	12085
12088	12091	12094	12097	12100	12103	12106	12109	12112	12115	12118	12121	12124	12127	12130
12133	12136	12139	12142	12145	12148	12151	12154	12157	12160	12163	12166	12169	12172	12175
12178	12181	12184	12187	12190	12193	12196	12199	12202	12205	12208	12211	12214	12217	12220
12223	12226	12229	12232	12235	12238	12241	12244	12247	12250	12253	12256	12259	12262	12265
12268	12271	12274	12277	12280	12283	12286	12289	12292	12295	12298	12301	12304	12307	12310
12313	12316	12319	12322	12325	12328	12331	12334	12337	12340	12343	12346	12349	12352	12355
12358	12361	12364	12367	12370	12373	12376	12379	12382	12385	12388	12391	12394	12397	12400
12403	12406	12409	12412	12415	12418	12421	12424	12427	12430	12433	12436	12439	12442	12445
12448	12451	12454	12457	12460	12463	12466	12469	12472	12475	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
12583	12586	12589	12592	12595	12598	12601	12604	12607	12610	12613	12616	12619	12622	12625
12628	12631	12634	12637	12640	12643	12646	12649	12652	12655	12658	12661	12664	12667	12670
12673	12676	12679	12682	12685	12688	12691	12694	12697	12700	12703	12706	12709	12712	12715
12718	12721	12724	12727	12730	12733	12736	12739	12742	12745	12748	12751	12754	12757	12760
12763	12766	12769	12772	12775	12778	12781	12784	12787	12790	12793	12796	12799	12802	12805
12808	12811	12814	12817	12820	12823	12826	12829	12832	12835	12838	12841	12844	12847	12850
12853	12856	12859	12862	12865	12868	12871	12874	12877	12880	12883	12886	12889	12892	12895
12898	12901	12904	12907	12910	12913	12916	12919	12922	12925	12928	12931	12934	12937	12940
12943	12946	12949	12952	12955	12958	12961	12964	12967	12970	12973	12976	12979	12982	12985
12988	12991	12994	12997	13000	13003	13006	13009	13012	13015	13018	13021	13024	13027	13030
13033	13036	13039	13042	13045	13048	13051	13054	13057	13060	13063	13066	13069	13072	13075
13078	13081	13084	13087	13090	13093	13096	13099	13102	13105	13108	13111	13114	13117	13120
13123	13126	13129	13132	13135	13138	13141	13144	13147	13150	13153	13156	13159	13162	13165
13168	13171	13174	13177	13180	13183	13186	13189	13192	13195	13198	13201	13204	13207	13210
13213	13216	13219	13222	13225	13228	13231	13234	13237	13240	13243	13246	13249	13252	13255
13258	13261	13264	13267	13270	13273	13276	13279	13282	13285	13288	13291	13294	13297	13300
13303	13306	13309	13312	13315	13318	13321	13324	13327	13330	13333	13336	13339	13342	13345
13348	13351	13354	13357	13360	13363	13366	13369	13372	13375	13378	13381	13384	13387	13390
13393	13396	13399	13402	13405	13408	13411	13414	13417	13420	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	13513	13516	13519	13522	13525
13528	13531	13534	13537	13540	13543	13546	13549	13552	13555	13558	13561	13564	13567	13570
13573	13576	13579	13582	13585	13588	13591	13594	13597	13600	13603	13606	13609	13612	13615
13618	13621	13624	13627	13630	13633	13636	13639	13642	13645	13648	13651	13654	13657	13660
13663	13666	13669	13672	13675	13678	13681	13684	13687	13690	13693	13696	13699	13702	13705
13708	13711	13714	13717	13720	13723	13726	13729	13732	13735	13738	13741	13744	13747	13750
13753	13756	13759	13762	13765	13768	13771	13774	13777	13780	13783	13786	13789	13792	13795
13798	13801	13804	13807	13810	13813	13816	13819	13822	13825	13828	13831	13834	13837	13840
13843	13846	13849	13852	13855	13858	13861	13864	13867	13870	13873	13876	13879	13882	13885
13888	13891	13894	13897	1										

CROSS REFERENCE TABLE -- MACRO NAMES

.DESTM

13962	13971	13974	13980	13990	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	14461	14484	14487	14490	14493	14496	14499	14502	14505	14508
14511	14514	14517	14520	14523	14526	14529	14536	14536	14538	14541	14544	14550	14553	14556
14562	14565	14568	14574	14580	14583	14589	14596	14598	14604	14607	14613	14623	14626	14632
14638	14641	14644	14647	14650	14654	14664	14669	14702	14705	14708	14711	14714	14717	14720
14723	14726	14729	14732	14738	14747	14750	14753	14758	14764	14815	14818	14821	14824	14827
14830	14833	14836	14839	14842	14845	14851	14860	14863	14866	14869	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	2297	2300	2312	2315	2318	2321	2324	2327	2331	2337
2344	2350	2356	2370	2374	2377	2380	2392	2395	2398	2401	2404	2407	2411	2417
2424	2430	2436	2447	2449	2463	2467	2491	2494	2497	2509	2512	2515	2518	2521
2524	2528	2534	2541	2547	2553	2567	2571	2574	2577	2589	2592	2595	2598	2601
2604	2608	2614	2621	2627	2633	2644	2677	2680	2694	2698	2691	2705	2709	2715
2712	2715	2718	2721	2725	2731	2738	2744	2750	2764	2768	2771	2774	2786	2789
2792	2795	2798	2801	2805	2811	2818	2824	2830	2841	2874	2877	2881	2885	2888
2891	2903	2906	2909	2912	2915	2918	2922	2928	2935	2941	2947	2961	2965	2968
2971	2983	2986	2989	2992	2995	2998	3002	3008	3015	3021	3027	3038	3072	3075
3079	3083	3086	3089	3095	3098	3110	3113	3116	3119	3122	3125	3129	3135	3142
3148	3154	3168	3172	3175	3178	3184	3187	3192	3202	3205	3208	3211	3214	3218
3224	3231	3237	3243	3254	3269	3272	3275	3278	3303	3306	3312	3315	3318	3321
3336	3339	3342	3345	3348	3351	3357	3360	3374	3449	3453	3459	3466	3472	3478
3410	3413	3416	3419	3434	3437	3440	3443	3449	3457	3460	3463	3466	3470	3476
3489	3522	3525	3529	3533	3537	3540	3543	3557	3577	3580	3583	3586	3590	3596
3583	3589	3595	3609	3613	3619	3626	3637	3640	3643	3646	3649	3650	3655	3656
3663	3669	3675	3686	3719	3786	3796	3816	3828	3831	3834	3837	3846	3850	3856
3863	3869	3875	3886	3896	3964	3966	3973	3983	3987	3990	3993	3996	3999	4002
3993	3994	3997	4004	4004	4004	4007	4007	4007	4007	4010	4013	4025	4028	4028
4011	4014	4017	4020	4023	4026	4029	4032	4035	4038	4041	4044	4047	4050	4053
4056	4059	4062	4065	4068	4071	4074	4077	4080	4113	4116	4120	4124	4127	4127
4130	4133	4136	4139	4142	4145	4148	4151	4154	4180	4186	4200	4204	4207	4213
4216	4219	4222	4225	4228	4231	4234	4237	4240	4260	4266	4277	4310	4313	4313
4243	4246	4249	4252	4255	4258	4261	4264	4267	4290	4296	4307	4377	4383	4383
4270	4273	4276	4279	4282	4285	4288	4291	4294	4328	4334	4345	4451	4463	4463
4300	4303	4306	4309	4312	4315	4318	4321	4324	4358	4364	4371	4457	4561	4561
4327	4330	4333	4336	4339	4342	4345	4348	4351	4385	4391	4397	4457	4561	4561
4354	4357	4360	4363	4366	4369	4372	4375	4378	4412	4418	4424	4457	4561	4561
4381	4384	4387	4390	4393	4396	4399	4402	4405	4439	4445	4451	4457	4561	4561
4412	4415	4418	4421	4424	4427	4430	4433	4436	4470	4476	4482	4488	4561	4561
4443	4446	4449	4452	4455	4458	4461	4464	4467	4501	4507	4513	4519	4561	4561
4474	4477	4480	4483	4486	4489	4492	4495	4498	4532	4538	4544	4550	4561	4561
4509	4512	4515	4518	4521	4524	4527	4530	4533	4567	4573	4579	4585	4591	4591
4544	4547	4550	4553	4556	4559	4562	4565	4568	4602	4608	4614	4620	4626	4626
4579	4582	4585	4588	4591	4594	4597	4600	4603	4637	4643	4649	4655	4661	4661
4618	4621	4624	4627	4630	4633	4636	4639	4642	4676	4682	4688	4694	4700	4700
4655	4658	4661	4664	4667	4670	4673	4676	4679	4713	4719	4725	4731	4737	4737
4694	4697	4700	4703	4706	4709	4712	4715	4718	4752	4758	4764	4770	4776	4776
4730	4733	4736	4739	4742	4745	4748	4751	4754	4788	4794	4800	4806	4812	4812
4765	4768	4771	4774	4777	4780	4783	4786	4789	4823	4829	4835	4841	4847	4847
4792	4795	4798	4801	4804	4807	4810	4813	4816	4850	4856	4862	4868	4874	4874
4809	4812	4815	4818	4821	4824	4827	4830	4833	4867	4873	4879	4885	4891	4891
4836	4839	4842	4845	4848	4851	4854	4857	4860	4894	4900	4906	4912	4918	4918
4863	4866	4869	4872	4875	4878	4881	4884	4887	4921	4927	4933	4939	4945	4945
4890	4893	4896	4899	4902	4905	4908	4911	4914	4948	4954	4960	4966	4972	4972
4905	4908	4911	4914	4917	4920	4923	4926	4929	4963	4969	4975	4981	4987	4987
4922	4925	4928	4931	4934	4937	4940	4943	4946	4980	4986	4992	4998	5004	5004
4949	4952	4955	4958	4961	4964	4967	4970	4973	5007	5013	5019	5025	5031	5031
4976	4979	4982	4985	4988	4991	4994	4997	5000	5034	5040	5046	5052	5058	5058
5003	5006	5009	5012	5015	5018	5021	5024	5027	5061	5067	5073	5079	5085	5085
5030	5033	5036	5039	5042	5045	5048	5051	5054	5088	5094	5100	5106	5112	5112
5057	5060	5063	5066	5069	5072	5075	5078	5081	5115	5121	5127	5133	5139	5139
5084	5087	5090	5093	5096	5099	5102	5105	5108	5142	5148	5154	5160	5166	5166
5101	5104	5107	5110	5113	5116	5119	5122	5125	5159	5165	5171	5177	5183	5183
5114	5117	5120	5123	5126	5129	5132	5135	5138	5172	5178	5184	5190	5196	5196
5131	5134	5137	5140	5143	5146	5149	5152	5155	5189	5195	5201	5207	5213	5213
5144	5147	5150	5153	5156	5159	5162	5165	5168	5202	5208	5214	5220	5226	5226
5161	5164	5167	5170	5173	5176	5179	5182	5185	5219	5225	5231	5237	5243	5243
5174	5177	5180	5183	5186	5189	5192	5195	5198	5232	5238	5244	5250	5256	5256
5181	5184	5187	5190	5193	5196	5199	5202	5205	5239	5245	5251	5257	5263	5263
5188	5191	5194	5197	5200	5203	5206	5209	5212	5246	5252	5258	5264	5270	5270
5195	5198	5201	5204	5207	5210	5213	5216	5219	5253	5259	5265	5271	5277	5277
5202	5205	5208	5211	5214	5217	5220	5223	5226	5260	5266	5272	5278	5284	5284
5209	5212	5215	5218	5221	5224	5227	5230	5233	5267	5273	5279	5285	5291	5291
5216	5219	5222	5225	5228	5231	5234	5237	5240	5274	5280	5286	5292	5298	5298
5223	5226	5229	5232	5235	5238	5241	5244	5247	5281	5287	5293	5299	5305	5305
5230	5233	5236	5239	5242	5245	5248	5251	5254	5288	5294	5300	5306	5312	5312
5237	5240	5243	5246	5249	5252	5255	5258	5261	5295	5301	5307	5313	5319	5319
5244	5247	5250	5253	5256	5259	5262	5265	5268	5302	5308	5314	5320	5326	5326
5251	5254	5257	5260	5263	5266	5269	5272	5275	5306	5312	5318	5324	5330	5330
5258	5261	5264	5267	5270	5273	5276	5279	5282	5310	5316	5322	5328	5334	

DZKCA.P11 13-MAY-77 13:58 CROSS REFERENCE TABLE -- MACRO NAMES

5910	5913	5916	5928	5931	5934	5937	5940	5943	5946	5950	5956	5963	5969	5975
5999	5998	5996	5998	6001	6013	6016	6019	6022	6025	6028	6031	6035	6041	6048
6054	6060	6071	6104	6107	6111	6114	6117	6120	6123	6135	6138	6141	6144	6147
6150	6153	6157	6163	6170	6176	6182	6196	6199	6202	6208	6208	6220	6223	6226
6229	6233	6235	6238	6242	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	6596	6613	6616	6619	6622	6634	6637	6640	6643
6646	6649	6652	6656	6662	6669	6675	6681	6692	6695	6725	6728	6732	6738	6741
6744	6756	6759	6762	6765	6768	6771	6774	6778	6784	6789	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	6956	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	7845	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	8050	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	8536	8552	8555	8558	8561	8564	8567	8579
8582	8585	8588	8591	8595	8605	8620	8627	8639	8669	8672	8678	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	8961	8964	8967	8970	8973	8976	F 379	8985	8988
8991	8994	8997	9000	9003	9006	9009	9012	9015	9018	9021	9024	127	9030	9033
9036	9039	9042	9045	9048	9051	9054	9057	9060	9063	9066	9069	9072	9075	9078
9081	9084	9088	9092	9098	9101	9104	9116	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	9595	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	10064	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	10356
10360	10363	10366	10369	10372	10375	10378	10381	10384	10387	10390	10393	10396	10399	10402
10405	10408	10411	10414	10417	10420	10423	10426	10429	10432	10435	10438	10441	10444	10447
10450	10453	10456	10459	10462	10465	10468	10471	10474	10477	10480	10483	10486	10489	10492
10495	10498	10501	10504	10507	10510	10513	10516	10519	10522	10525	10528	10531	10534	10537
10540	10543	10546	10549	10552	10555	10558	10561	10564	10567	10570	10573	10576	10579	10582
10585	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	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	10801	10804	10807
10810	10813	10816	10819	10822	10825	10828	10831	10834	10837	10840	10843	10846	10849	10852
10855	10858	10861	10864	10867	10870	10873	10876	10879	10882	10885	10888	10891	10894	10897
10899	10902	10905	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	10998	11001	11004	11007	11010	11013	11016	11019	11022	11025	11028	11031
11034	11037	11040	11043	11046	11049	11052	11055	11058	11061	11064	11067	11070	11073	11076
11079	11082	11085	11088	11091	11094	11097	11100	11103	11106	11109	11112	11115	11118	11121
11124	11127	11130	11133	11136	11139	11142	11145	11148	11151	11154	11157	11160	11163	11166
11169	11172	11175	11178	11181	11184	11187	11190	11193	11196	11199	11202	11205	11208	11211
11214	11217	11220	11223	11226	11229	11232	11235	11238	11241	11244	11247	11250	11253	11256
11259	11262	11265	11268	11271	11274	11277	11280	11283	11286	11289	11292	11295	11298	11301
11304	11307	11310	11313	11316	11319	11322	11325	11328	11331	11334	11337	11340	11343	11346
11349	11352	11355	11358	11361	11364	11367	11370	11373	11376	11379	11382	11385	11388	11391
11394	11397	11400	11403	11406	11409	11412	11415	11418	11421	11424	11427	11430	11433	11436
11439	11442	11445	11448	11451	11454	11457	11460	11463	11466	11469	11472	11475	11478	11481
11484	11487	11490	11493	11496	11499	11502	11505	11508	11511	11514	11517	11520	11523	11526
11529	11532	11535	11538	11541	11544	11547	11550	11553	11556	11559	11562	11565	11568	11571
11574	11577	11580	11583	11586	11589	11592	11595	11598	11601	11604	11607	11610	11613	11616
11619	11622	11625	11628	11631	11634	11637	11640	11643	11646	11649	11652	11655	11658	11661
11664	11667	11670	11673	11676	11679	11682	11685	11688	11691	11694	11697	11700	11703	11706
11709	11712	11715	11718	11721	11724	11727	11730	11733	11736	11739	11742	11745	11748	11751
11754	11757	11760	11763	11766	11769	11772	11775	11778	11781	11784	11787	11790	11793	11796
11799	11802	11805	11808	11811	11814	11817	11820	11823	11826	11829	11832	11835	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	11973	11976
11979	11982	11985	11988	11991	11994	11997	12000	12003	12006	12009	12012	12015	12018	12021
12024	12027	12030	12033	12036	12039	12042	12045	12048	12051	12054	12057	12060	12063	12066
12069	12072	12075	12078	12081	12084	12087	12090	12093	12096	12099	12102	12105	12108	12111
12114	12117	12120	12123	12126	12129	12132	12135	12138	12141	12144	12147	12150	12153	12156
12159	12162	12165	12168	12171	12174	12177	12180	12183	12186	12189	12192	12195	12198	12201
12204	12207	12210	12213	12216	12219	12222	12225	12228	12231	12234	12237	12240	12243	12246
12249	12252	12255	12258	12261	12264	12267	12270	12273	12276	12279	12282	12285	12288	12291
12294	12297	12300	12303	12306	12309	12312	12315	12318	12321	12324	12327	12330	12333	12336
12339	12342	12345	12348	12351	12354	12357	12360	12363	12366	12369	12372	12375	12378	12381
12384	12387	12390	12393	12396	12399	12402	12405	12408	12411	12414	12417	12420	12423	12426
12429	12432	12435	12438	12441	12444	12447	12450	12453	12456	12459	12462	12465	12468	12471
12474	12477	12480	12483	12486	12489	12492	12495	12498	12501	12504	12507	12510	12513	12516
12519	12522	12525	12528	12531	12534	12537	12540	12543	12546	12549	12552	12555	12558	12561
12564	12567	12570	12573	12576	12579	12582	12585	12588	12591	12594	12597	12600	12603	12606
12609	12612	12615	12618	12621	12624	12627	12630	12633	12636	12639	12642	12645	12648	12651
12654	12657	12660	12663	12666	12669	12672	12675	12678	12681	12684	12687	12690	12693	12696
12699	12702	12705	12708	12711	12714	12717	12720	12723	12726	12729	12732	12735	12738	12741
12744	12747	12750	12753	12756	12759	12762	12765	12768	12771	12774	12777	12780	12783	12786
12789	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	12909	12912	12915	12918	12921
12924	12927	12930	12933	12936	12939	12942	12945	12948	12951	12954	12957	12960	12963	12966
12969	12972	12975	12978	12981	12984	12987	12990	12993	12996	12999	13002	13005	13008	13011
13014	13017	13020	13023	13026	13029	13032	13035	13038	13041	13044	13047	13050	13053	13056
13059	13062	13065	13068	13071	13074	13077	13080	13083	13086	13089	13092	13095	13098	13101
13104	13107	13110	13113	13116	13119	13122	13125	13128	13131	13134	13137	13140	13143	13146
13149	13152	13155	13158	13161	13164	13167	13170	13173	13176	13179	13182	13185	13188	13191
13194	13197	13200	13203	13206	13209	13212	13215	13218	13221	13224	13227	13230	13233	13236
13239	13242	13245	13248	13251	13254	13257	13260	13263	13266	13269	13272	13275	13278	13281
13284	13287	13290	13293	13296	13299	13302	13305	13308	13311	13314	13317	13320	13323	13326
13329	13332	13335	13338	13341	13344	13347	13350	13353	13356	13359	13362	13365	13368	13371
13374	13377	13380	13383	13386	13389	13392	13395	13398	13401	13404	13407	13410	13413	13416
13419	13422	13425	13428	13431	13434	13437	13440	13443	13446	13449	13452	13455	13458	13461
13464	13467	13470	13473	13476	13479	13482	13485	13488	13491	13494	13497	13500	13503	13506
13509	13512	13515	13518	13521	13524	13527	13530	13533	13536	13539	13542	13545	13548	13551
13554	13557	13560	13563	13566	13569	13572	13575	13578	13581	13584	13587	13590	13593	13596
13599	13602	13605	13608	13611	13614	13617	13620	13623	13626	13629	13632	13635	13638	13641
13644	13647	13650	13653	13656	13659	13662	13665	13668	13671	13674	13677	13680	13683	13686
13689	13692	13695	13698	13701	13704	13707	13710	13713	13716	13719	13722	13725	13728	13731
13734	13737	13740	13743	13746	13749	13752	13755	13758	13761	13764	13767	13770	13773	13776
13779														

DZKCA.P11 13-MAY-77 13:58

CROSS REFERENCE TABLE -- MACRO NAMES

14141	14145	14151	14155	14161	14165	14171	14175	14181	14185	14191	14195	14201	14207	14211
14217	14223	14229	14235	14241	14247	14253	14259	14265	14271	14277	14283	14289	14295	14301
14297	14303	14309	14315	14321	14327	14333	14339	14345	14351	14357	14363	14369	14375	14381
14387	14393	14399	14405	14411	14417	14423	14429	14435	14441	14447	14453	14459	14465	14471
14477	14483	14489	14495	14501	14507	14513	14519	14525	14531	14537	14543	14549	14555	14561
14567	14573	14579	14585	14591	14597	14603	14609	14615	14621	14627	14633	14639	14645	14651
14657	14663	14669	14675	14681	14687	14693	14699	14705	14711	14717	14723	14729	14735	14741
14747	14753	14759	14765	14771	14777	14783	14789	14795	14801	14807	14813	14819	14825	14831
14837	14843	14849	14855	14861	14867	14873	14879	14885	14891	14897	14903	14909	14915	14921
14927	14933	14939	14945	14951	14957	14963	14969	14975	14981	14987	14993	14999	15005	15011
15017	15023	15029	15035	15041	15047	15053	15059	15065	15071	15077	15083	15089	15095	15101
15107	15113	15119	15125	15131	15137	15143	15149	15155	15161	15167	15173	15179	15185	15191
15197	15203	15209	15215	15221	15227	15233	15239	15245	15251	15257	15263	15269	15275	15281
15287	15293	15299	15305	15311	15317	15323	15329	15335	15341	15347	15353	15359	15365	15371
15377	15383	15389	15395	15401	15407	15413	15419	15425	15431	15437	15443	15449	15455	15461
15467	15473	15479	15485	15491	15497	15503	15509	15515	15521	15527	15533	15539	15545	15551
15557	15563	15569	15575	15581	15587	15593	15599	15605	15611	15617	15623	15629	15635	15641
15647	15653	15659	15665	15671	15677	15683	15689	15695	15701	15707	15713	15719	15725	15731
15737	15743	15749	15755	15761	15767	15773	15779	15785	15791	15797	15803	15809	15815	15821
15827	15833	15839	15845	15851	15857	15863	15869	15875	15881	15887	15893	15899	15905	15911
15917	15923	15929	15935	15941	15947	15953	15959	15965	15971	15977	15983	15989	15995	16001
16007	16013	16019	16025	16031	16037	16043	16049	16055	16061	16067	16073	16079	16085	16091
16097	16103	16109	16115	16121	16127	16133	16139	16145	16151	16157	16163	16169	16175	16181
16187	16193	16199	16205	16211	16217	16223	16229	16235	16241	16247	16253	16259	16265	16271
16277	16283	16289	16295	16301	16307	16313	16319	16325	16331	16337	16343	16349	16355	16361
16367	16373	16379	16385	16391	16397	16403	16409	16415	16421	16427	16433	16439	16445	16451
16457	16463	16469	16475	16481	16487	16493	16499	16505	16511	16517	16523	16529	16535	16541
16547	16553	16559	16565	16571	16577	16583	16589	16595	16601	16607	16613	16619	16625	16631
16637	16643	16649	16655	16661	16667	16673	16679	16685	16691	16697	16703	16709	16715	16721
16727	16733	16739	16745	16751	16757	16763	16769	16775	16781	16787	16793	16799	16805	16811
16817	16823	16829	16835	16841	16847	16853	16859	16865	16871	16877	16883	16889	16895	16901
16907	16913	16919	16925	16931	16937	16943	16949	16955	16961	16967	16973	16979	16985	16991
16997	17003	17009	17015	17021	17027	17033	17039	17045	17051	17057	17063	17069	17075	17081
17087	17093	17099	17105	17111	17117	17123	17129	17135	17141	17147	17153	17159	17165	17171
17177	17183	17189	17195	17201	17207	17213	17219	17225	17231	17237	17243	17249	17255	17261
17267	17273	17279	17285	17291	17297	17303	17309	17315	17321	17327	17333	17339	17345	17351
17357	17363	17369	17375	17381	17387	17393	17399	17405	17411	17417	17423	17429	17435	17441
17447	17453	17459	17465	17471	17477	17483	17489	17495	17501	17507	17513	17519	17525	17531
17537	17543	17549	17555	17561	17567	17573	17579	17585	17591	17597	17603	17609	17615	17621
17627	17633	17639	17645	17651	17657	17663	17669	17675	17681	17687	17693	17699	17705	17711
17717	17723	17729	17735	17741	17747	17753	17759	17765	17771	17777	17783	17789	17795	17801
17807	17813	17819	17825	17831	17837	17843	17849	17855	17861	17867	17873	17879	17885	17891
17897	17903	17909	17915	17921	17927	17933	17939	17945	17951	17957	17963	17969	17975	17981
17987	17993	17999	18005	18011	18017	18023	18029	18035	18041	18047	18053	18059	18065	18071
18077	18083	18089	18095	18101	18107	18113	18119	18125	18131	18137	18143	18149	18155	18161
18167	18173	18179	18185	18191	18197	18203	18209	18215	18221	18227	18233	18239	18245	18251
18257	18263	18269	18275	18281	18287	18293	18299	18305	18311	18317	18323	18329	18335	18341
18347	18353	18359	18365	18371	18377	18383	18389	18395	18401	18407	18413	18419	18425	18431
18437	18443	18449	18455	18461	18467	18473	18479	18485	18491	18497	18503	18509	18515	18521
18527	18533	18539	18545	18551	18557	18563	18569	18575	18581	18587	18593	18599	18605	18611
18617	18623	18629	18635	18641	18647	18653	18659	18665	18671	18677	18683	18689	18695	18701
18707	18713	18719	18725	18731	18737	18743	18749	18755	18761	18767	18773	18779	18785	18791
18797	18803	18809	18815	18821	18827	18833	18839	18845	18851	18857	18863	18869	18875	18881
18887	18893	18899	18905	18911	18917	18923	18929	18935	18941	18947	18953	18959	18965	18971
18977	18983	18989	18995	19001	19007	19013	19019	19025	19031	19037	19043	19049	19055	19061
19067	19073	19079	19085	19091	19097	19103	19109	19115	19121	19127	19133	19139	19145	19151
19157	19163	19169	19175	19181	19187	19193	19199	19205	19211	19217	19223	19229	19235	19241
19247	19253	19259	19265	19271	19277	19283	19289	19295	19301	19307	19313	19319	19325	19331
19337	19343	19349	19355	19361	19367	19373	19379	19385	19391	19397	19403	19409	19415	19421
19427	19433	19439	19445	19451	19457	19463	19469	19475	19481	19487	19493	19499	19505	19511
19517	19523	19529	19535	19541	19547	19553	19559	19565	19571	19577	19583	19589	19595	19601
19607	19613	19619	19625	19631	19637	19643	19649	19655	19661	19667	19673	19679	19685	19691
19697	19703	19709	19715	19721	19727	19733	19739	19745	19751	19757	19763	19769	19775	19781
19787	19793	19799	19805	19811	19817	19823	19829	19835	19841	19847	19853	19859	19865	19871
19877	19883	19889	19895	19901	19907	19913	19919	19925	19931	19937	19943	19949	19955	19961
19967	19973	19979	19985	19991	19997	20003	20009	20015	20021	20027	20033	20039	20045	20051
20057	20063	20069	20075	20081	20087	20093	20099	20105	20111	20117	20123	20129	20135	20141
20147	20153	20159	20165	20171	20177	20183	20189	20195	20201	20207	20213	20219	20225	20231
20237	20243	20249	20255	20261	20267	20273	20279	20285	20291	20297	20303	20309	20315	20321
20327	20333	20339	20345	20351	20357	20363	20369	20375	20381	20387	20393	20399	20405	20411
20417	20423	20429	20435	20441	20447	20453	20459	20465	20471	20477	20483	20489	20495	20501
20507	20513	20519	20525	20531	20537	20543	20549	20555	20561	20567	20573	20579	20585	20591
20597	20603	20609	20615	20621	20627	20633	20639	20645	20651	20657	20663	20669	20675	20681
20687	20693	20699	20705	20711	20717	20723	20729	20735	20741	20747	20753	20759	20765	20771
20777	20783	20789	20795	20801	20807	20813	20819	20825	20831	20837	20843	20849	20855	20861
20867	20873	20879	20885	20891	20897	20903	20909	20915	20921	20927	20933	20939	20945	20951
20957	20963	20969	20975	20981	20987	20993	20999	21005	21011	21017	21023	21029	21035	21041
21047	21053	21059	21065	21071	21077	21083	21089	21095	21101	21107	21113	21119	21125	21131
21137														

CROSS REFERENCE TABLE -- MACRO NAMES

5700	5701	5702	5703	5704	5705	5706	5707	5708	5709	5710	5711	5712	5713	5714	5715	5716	5717	5718	5719	5720	5721	5722	5723	5724	5725	5726	5727	5728	5729	5730	5731	5732	5733	5734	5735	5736	5737	5738	5739	5740	5741	5742	5743	5744	5745	5746	5747	5748	5749	5750	5751	5752	5753	5754	5755	5756	5757	5758	5759	5760	5761	5762	5763	5764	5765	5766	5767	5768	5769	5770	5771	5772	5773	5774	5775	5776	5777	5778	5779	5780	5781	5782	5783	5784	5785	5786	5787	5788	5789	5790	5791	5792	5793	5794	5795	5796	5797	5798	5799	5800	5801	5802	5803	5804	5805	5806	5807	5808	5809	5810	5811	5812	5813	5814	5815	5816	5817	5818	5819	5820	5821	5822	5823	5824	5825	5826	5827	5828	5829	5830	5831	5832	5833	5834	5835	5836	5837	5838	5839	5840	5841	5842	5843	5844	5845	5846	5847	5848	5849	5850	5851	5852	5853	5854	5855	5856	5857	5858	5859	5860	5861	5862	5863	5864	5865	5866	5867	5868	5869	5870	5871	5872	5873	5874	5875	5876	5877	5878	5879	5880	5881	5882	5883	5884	5885	5886	5887	5888	5889	5890	5891	5892	5893	5894	5895	5896	5897	5898	5899	5900	5901	5902	5903	5904	5905	5906	5907	5908	5909	5910	5911	5912	5913	5914	5915	5916	5917	5918	5919	5920	5921	5922	5923	5924	5925	5926	5927	5928	5929	5930	5931	5932	5933	5934	5935	5936	5937	5938	5939	5940	5941	5942	5943	5944	5945	5946	5947	5948	5949	5950	5951	5952	5953	5954	5955	5956	5957	5958	5959	5960	5961	5962	5963	5964	5965	5966	5967	5968	5969	5970	5971	5972	5973	5974	5975	5976	5977	5978	5979	5980	5981	5982	5983	5984	5985	5986	5987	5988	5989	5990	5991	5992	5993	5994	5995	5996	5997	5998	5999	6000
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

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	10120	10123	10126	10135	10138	10141	10144	10147	10150	10154	10157	10160
10173	10179	10194	10229	10232	10235	10238	10241	10244	10247	10250	10253	10257	10260	10263
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	10464	10467	10471	10477	10484
10488	10511	10516	10519	10522	10525	10528	10531	10534	10537	10540	10543	10549	10552	10555
10558	10591	10596	10599	10602	10605	10608	10611	10614	10617	10620	10624	10628	10634	10637
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	10723	10726
10748	10754	10757	10760	10763	10766	10769	10772	10775	10778	10781	10784	10787	10791	10794
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	10994	11000
11003	11009	11015	11018	11021	11024	11027	11030	11034	11040	11047	11053	11059	11062	11065
11071	11074	11080	11083	11086	11089	11092	11095	11098	11105	11111	11118	11124	11130	11145
11180	11183	11186	11189	11192	11195	11198	11201	11204	11207	11210	11213	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	11307	11311	11317	11320
11326	11332	11335	11338	11341	11344	11347	11350	11353	11356	11364	11370	11376	11379	11388
11391	11397	11403	11406	11409	11412	11415	11418	11421	11428	11438	11441	11447	11462	11497
11500	11503	11506	11509	11512	11515	11518	11521	11524	11527	11530	11536	11539	11542	11545
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	11624	11628	11634	11637	11643
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	12006	12013	12016	12022
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
12283	12286	12289	12292	12295	12298	12301	12304	12307	12310	12313	12316	12319	12322	12325
12333	12336	12339	12342	12345	12348	12351	12354	12357	12360	12363	12366	12369	12372	12375
12383	12386	12389	12392	12395	12398	12401	12404	12407	12410	12413	12416	12419	12422	12425
12433	12436	12439	12442	12445	12448	12451	12454	12457	12460	12463	12466	12469	12472	12475
12483	12486	12489	12492	12495	12498	12501	12504	12507	12510	12513	12516	12519	12522	12525
12533	12536	12539	12542	12545	12548	12551	12554	12557	12560	12563	12566	12569	12572	12575
12583	12586	12589	12592	12595	12598	12601	12604	12607	12610	12613	12616	12619	12622	12625
12633	12636	12639	12642	12645	12648	12651	12654	12657	12660	12663	12666	12669	12672	12675
12683	12686	12689	12692	12695	12698	12701	12704	12707	12710	12713	12716	12719	12722	12725
12733	12736	12739	12742	12745	12748	12751	12754	12757	12760	12763	12766	12769	12772	12775
12783	12786	12789	12792	12795	12798	12801	12804	12807	12810	12813	12816	12819	12822	12825
12833	12836	12839	12842	12845	12848	12851	12854	12857	12860	12863	12866	12869	12872	12875
12883	12886	12889	12892	12895	12898	12901	12904	12907	12910	12913	12916	12919	12922	12925
12933	12936	12939	12942	12945	12948	12951	12954	12957	12960	12963	12966	12969	12972	12975
12983	12986	12989	12992	12995	12998	13001	13004	13007	13010	13013	13016	13019	13022	13025
13033	13036	13039	13042	13045	13048	13051	13054	13057	13060	13063	13066	13069	13072	13075
13083	13086	13089	13092	13095	13098	13101	13104	13107	13110	13113	13116	13119	13122	13125
13133	13136	13139	13142	13145	13148	13151	13154	13157	13160	13163	13166	13169	13172	13175
13183	13186	13189	13192	13195	13198	13201	13204	13207	13210	13213	13216	13219	13222	13225
13233	13236	13239	13242	13245	13248	13251	13254	13257	13260	13263	13266	13269	13272	13275
13283	13286	13289	13292	13295	13298	13301	13304	13307	13310	13313	13316	13319	13322	13325
13333	13336	13339	13342	13345	13348	13351	13354	13357	13360	13363	13366	13369	13372	13375
13383	13386	13389	13392	13395	13398	13401	13404	13407	13410	13413	13416	13419	13422	13425
13433	13436	13439	13442	13445	13448	13451	13454	13457	13460	13463	13466	13469	13472	13475
13483	13486	13489	13492	13495	13498	13501	13504	13507	13510	13513	13516	13519	13522	13525

N13

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

PAGE: 0376

DZKCA.P11 13-MAY-77 13:58

CROSS REFERENCE TABLE -- MACRO NAMES

.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: 55K (110 PAGES)