KMC11

CPU MICRO DIAGNOSTICS

MD-11-DZKCA-A

EP-DZKCA-A-DL-A

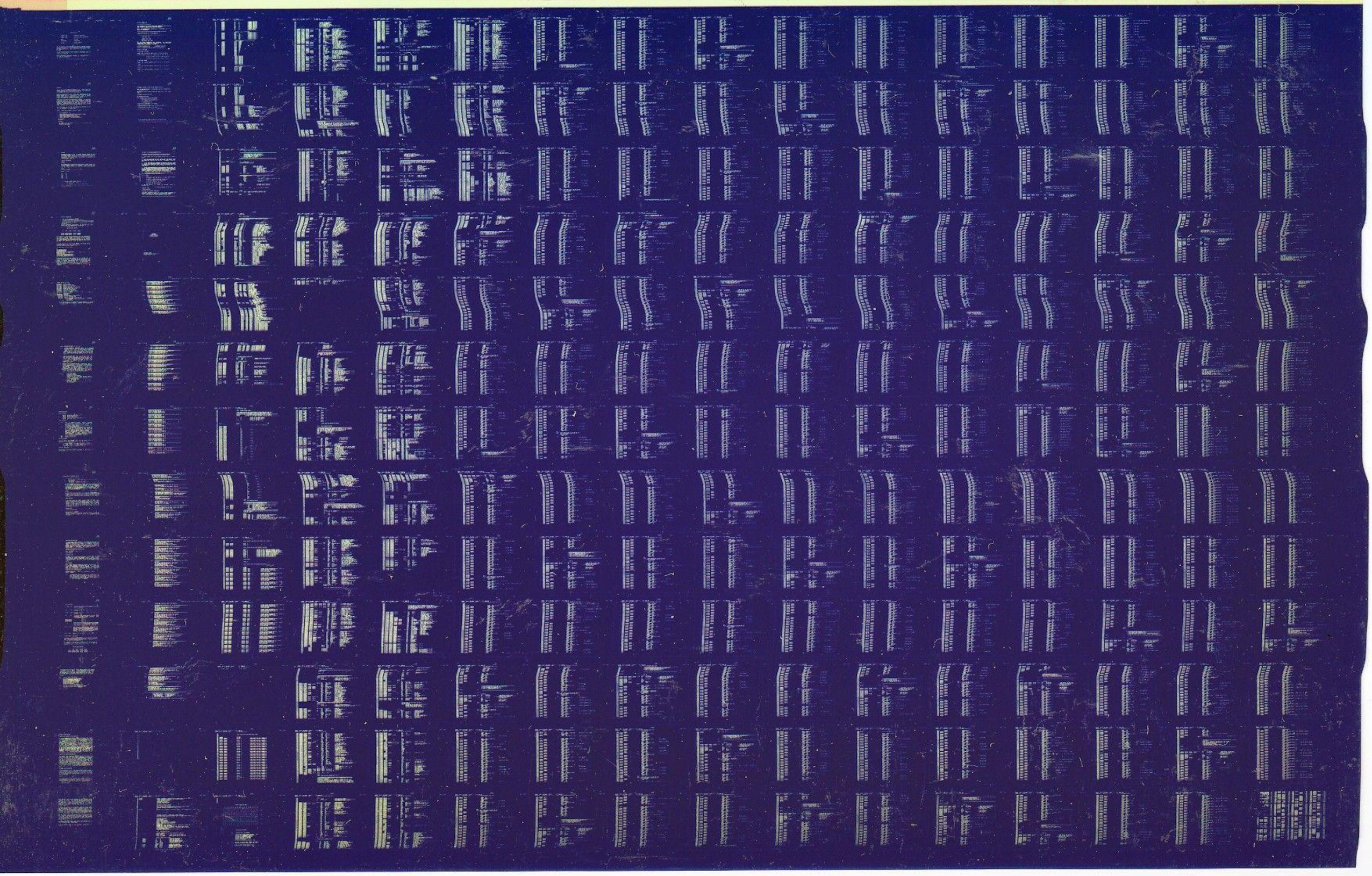
COPYRIGHT © 1977

FICHE 1 OF 2

AUG 1977

digital

MADE IN USA



KMC11

CPU MICRO DIAGNOSTICS

MD-11-DZKCA-A

EP-DZKCA-A-DL-A

COPYRIGHT © 1977

FICHE 2 OF 2

AUG 1977

digital

MADE IN USA

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 verfiy 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-processer (M8204). It performs write/read tests on the KMC unibus registers, checks the micro-processer operation, checks out Main Memory, scratch pad memory, the ALU functions as well as interrupts and NPR 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-processer 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 equilivalent)
KMC11-AN IOP (M8204)

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 PROCEEDURE

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

MAP OF KMC11 STATUS

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 (DDCMP LINE#)?377
SWITCH PAC#2 (BM873 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

SW 15 Set: Halt on error
SW 14 Set: Loop on current test
SW 13 Set: Inhibit error print out
SW 12 Set: Inhibit type out abell on error.
SW 11 Set: Inhibit iterations. (quick pass)
SW 10 Set: Escape to next test on error
SW 09 Set: Loop with current data
SW 08 Set: Catch error and loop on it
SW 07 Set: Use previous status table.
SW 06 Set: Halt in ROMCLK routine before clocking micro-processor
SW 05 Set: Reserved
SW 04 Set: Reserved
SW 03 Set: Reselect KMC11's desired active
SW 02 Set: Lock on selected test
SW 01 Set: Restart program at selected test
SW 00 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 carrige return to begin execution at the selected test.

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

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

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

4.1.3 DYNAMIC SWITCHES

ERROR SWITCHES

1. SW 12 Delete print out/bell on error.
2. SW 13 Delete error printout.
3. SW 15 Halt on the error.
4. SW 08 Goto beginning of the test(on error).
5. SW 10 Goto 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 incerporated 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 intermittemt errors; SW14=1 will loop on test reguardless of error or not error. (SW14=1, SW10=0, SW09=0, SW08=1,0)
3. SW11 Inhibit interations.
4. SW14 Loop on current test.

4.2 STARTING ADDRESS

Starting address is at 000200 there are no other starting addresses for the KMC11 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 KMC11's are tested the program will return to 'XXDP' or 'ACT-11'.

5. OPERATING PROCEDURE

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

5.2 PROGRAM AND/OR OPERATOR ACTION

The typical approach should be

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

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

6. ERRORS

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

6.2 ERROR RECOVERY

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

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

See section 4. (PLEASE)
Status table should be verified reguardless of how program was started. Also it is important to use this listing along with the information printed on the TTY to completly 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 (SW00=1) or by autosizing (SW00=0 and SW07=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 SW07=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 cummulitive 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 peformed.

STSTNM (1202) Contains the number of the test now being peformed.

RUN (1500) The bit in 'RUN' always points to the KMC11 currently being tested. EXAMPLE: (RUN) 1500/0000000001000000 Means that KMC11 no.06 is the KMC11 now running.

KMCR00-KMCR17
KMST00-KMST17
(2100)-(2300)

These locations contain the information needed to test up to 16 (decimal) KMC11s sequentialy. 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) 1470/0000000000011111 means that KMC11 no. 00,01,02,03,04 will be tested. EXAMPLE: (KMACTV) 1470/0000000000010001 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 CRAM, and a 125252 indicates a KMC11 with CRAM. 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 interupt occures 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 interupt occured; the address to which the KMC11 interupted 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.

8.6 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 (SWR = XXXXXX NEW?).

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

APT/ACT/XXDP/SLIDE

++++++++++++++++++++++++++

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

R0=POINTS TO GOOD MICRO-CODE INSTRUCTION.

R1=HAS CSR ADDRESS OF KMC UNDER TEST.

R2=POINTS THE CRAM LOCATION.

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

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

" ERROR MESSAGE "

GOOD BAD UPC XXXXXX[IF REQUIRED]

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

****
*

DOCUMENT
****
MAINDEC-11-DZKCA-A
****

COPYRIGHT 1977
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS. 01754

```
2262    ************************** TEST 1 ****************************
         MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
         FLOAT A 1 THROUGH REGISTER OUT1 (4)
         FLOAT A 0THROUGH REGISTER OUT1 (4)

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

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

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

3050    ************************** TEST 5 ****************************
         MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
         FLOAT A 1 THROUGH REGISTER OUT1 (10)
         FLOAT A 0THROUGH REGISTER OUT1 (10)
         THE NPR RQ 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 0THROUGH REGISTER OUT1 (11)
         THE BR RQ 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 0THROUGH REGISTER OUT0 (0)
```

```
3698    ********** TEST 10 **********
         MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.

3700     FLOAT A 1 THROUGH REGISTER OUTO (1)
         FLOAT A 0THROUGH REGISTER OUTO (1)

3895    ********** TEST 11 **********
         MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
         FLOAT A 1 THROUGH REGISTER OUTO (2)
         FLOAT A 0THROUGH REGISTER OUTO (2)

4092    ********** TEST 12 **********
         MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
         FLOAT A 1 THROUGH REGISTER OUTO (3)
         FLOAT A 0THROUGH REGISTER OUTO (3)

4289    ********** TEST 13 **********
         MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
         FLOAT A 1 THROUGH REGISTER OUTO (4)
         FLOAT A 0THROUGH REGISTER OUTO (4)

4486    ********** TEST 14 **********
         MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
         FLOAT A 1 THROUGH REGISTER OUTO (5)
         FLOAT A 0THROUGH REGISTER OUTO (5)

4683    ********** TEST 15 **********
         MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
         FLOAT A 1 THROUGH REGISTER OUTO (6)
         FLOAT A 0THROUGH REGISTER OUTO (6)

4880    ********** TEST 16 **********
         MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
         FLOAT A 1 THROUGH REGISTER OUTO (7)
         FLOAT A 0THROUGH 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 **********************
        MPR 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 **********************
        MPR TEST
        TEST OF MULTIPLE MPR'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      NCTION A OR B WITH C BIT SET.
        ALU FUNC      )
        LOAD MA      WORDS OF DATA.
        PERFORM       VERIFY THE RESULTS..

10206   ********      ***** TEST 47 ***************************
        ALU TES
        TEST OF       A XOR B WITH C BIT CLEARED.
        TEST OF       A XOR B WITH C BIT SET.
        ALU FUNC
        LOAD MAI      OF DATA.
        PERFORM       / THE RESULTS..

10523   ********      ***  TEST 48 ***************************
        ALU TEST
        TEST OF            C CLEARED.
        TEST OF
        ALU FUNC
        LOAD MAI
        PERFORM                    TS..

10840   *********                  ************************
        ALU TEST
        TEST OF                    CLEARED.
        TEST OF                    ET.
        ALU FUNC
        LOAD MAI
        PERFORM

11157   *******                    ***************
        ALU TEST
        TEST OF  A+B+C
        TEST OF  INCTI.
        ALU FUNC A+B+
        LOAD MAIN    RY 16 WORDS OF DAT
        PERFORM      NCTION  VERIFY THE R

11474   *********  ************ TEST 53 ***********************
        ALU TEST
        TEST OF ALU FUNCTION ADD A/C WITH C BIT CLEARED.
        TEST OF ALU FUNCTION ADD A/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 IBUS [illegible] SP,& MAIN MEMORY
        THEN GO [illegible] BACK THE DATA PATTERN
        TO VERI[illegible] READING AND WRITING OF OTHER
        LOCATIO[illegible] NOT CHANGE DATA.

14461   ********************** TEST 64 **********************
         HELL RAISER TEST...
         ONLY TO TEST MPRAM CONTROL LOGIC...
         NOT FOR MAINTENENCE 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 XXO.
                1) CSR4:=GOOD DATA      2) CSR5:=BAD DATA
                3) CSR3:=ERROR TYPE     4) CSR6:=ERROR PC IN MICRO-CODE.
                5) CSR7:=MISCALLANEOUS INFORMATION.

        ************************************************************
```

```
                        KMERNT  ↑?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-DZQAC-C3), JAN 19, 1977.
                        ;*
000002                  $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
                        ;SW8=0  AUTOSIZE KMC11
                        ;SW07=1  USE CURRENT KMC11 PARAMETERS
                        ;SW00=1  INPUT NEW KMC11 PARAMETERS
                        ;PRESS START
                        ;PROGRAM WILL TYPE "MAINDEC-11-DZKCA-A   BASIC KMC11 CONTROLLER TEST"
                        ;PROGRAM WILL TYPE STATUS MAP
                        ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
                        ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
                        ;AND THEN RESUME TESTING
                        ;SUBSEQUENT RESTARTS WILL NOT TYPE PROGRAM TITLE

                        .SBTTL  BASIC DEFINITIONS

                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1200 ***
001200                  STACK=  1200
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                        ;*MISCELLANEOUS DEFINITIONS
000011                  HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                  LF=     12              ;;CODE FOR LINE FEED
000015                  CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                  CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                  PS=     177776          ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
177774                  STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                  DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                  DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
```

```
                        ;*GENERAL REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
000006                  SP=     %6              ;;STACK POINTER
000007                  PC=     %7              ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
000002                  SW01=   2
000001                  SW00=   1
                        .EQUIV  SW00,SW
                        .EQUIV  SW01,SW1
                        .EQUIV  SW02,SW2
                        .EQUIV  SW03,SW3
                        .EQUIV  SW04,SW4
                        .EQUIV  SW05,SW5
                        .EQUIV  SW06,SW6
                        .EQUIV  SW07,SW7
                        .EQUIV  SW08,SW10
                        .EQUIV  SW09,SW11
                        .EQUIV  SW10,SW12
                        .EQUIV  SW11,SW13
                        .EQUIV  SW12,SW14
                        .EQUIV  SW13,SW15

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                  BIT15=  100000
040000                  BIT14=  40000
020000                  BIT13=  20000
010000                  BIT12=  10000
004000                  BIT11=  4000
002000                  BIT10=  2000
```

```
                        
                        BIT09=  1000
                        BIT08=  400
                        BIT07=  200
                        BIT06=  100
                        BIT05=  40
                        BIT04=  20
                        BIT03=  10
                        BIT02=  4
                        BIT01=  2
                        BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0
                        
                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                  TBITVEC=14              ;;"T" BIT
000014                  TRTVEC= 14              ;;TRACE TRAP
000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                  PWRVEC= 24              ;;POWER FAIL
000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                  TRAPVEC=34              ;;"TRAP" TRAP
000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
                        
                        
                        
                        ;INSTRUCTION DEFINITIONS
                        ;-----------------------
                        
005746                  PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD
005726                  POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD
010046                  PUSHR0=10046    ;SAVE R0 ON STACK
012600                  POPR0=12600     ;RESTORE R0 FROM STACK
024646                  PUSH2SP=24646   ;DECREMENT STACK TWICE
022626                  POP2SP=22626    ;INCREMENT STACK TWICE
                        .EQUIV  EMT,HLT ;BASIC DEFINITION OF ERROR CALL
                        
                        
                        
```

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

        000000                  .=0
000000  000000  000000                  .WORD   0,0
                                ;STANDARD INTERRUPT VECTORS
                                ;-------------------------

        000020                  .=20
000020  004134                          $SCOPE          ; SCOPE LOOP HANDLER.
000022  000340                          PR7             ; SERVICE AT LEVEL 7.
000024  007100                          $PWRDN                  ;POWER FAIL HANDLER
000026  000340                          PR7                     ;SERVICE AT LEVEL 7
000030  006502                          $ERROR                  ;ERROR HANDLER
000032  000340                          PR7                     ;SERVICE AT LEVEL 7
000034  006404                          $TRAP                   ;GENERAL HANDLER DISPATCH SERVICE
000036  000340                          PR7                     ;SERVICE AT LEVEL 7
                                .SBTTL  ACT11 HOOKS

                                ;;*****************************************************************
                                ;HOOKS REQUIRED BY ACT11
        000040                          $SVPC=.                 ;SAVE PC
        000046                          .=46
000046  004070                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF SENDAD IN .SEOP
        000052                          .=52
000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
        000040                          .=$SVPC                 ;; RESTORE PC

        000174                  .=174
000174  000000                  DISPREG:0                       ;SOFTWARE DISPLAY REGISTER
000176  000000                  SWREG:  0                       ;SOFTWARE SWITCH REGISTER

        000200                  .=200
000200  000137  002402                  JMP     .START          ;GO TO START OF PROGRAM


        001000                  .=1000
001000  005200  040515  047111  MTITLE: .ASCII  <200><12>/MAINDEC-11-DZKCA-A/<200>
(2) 001025     102  051501  041511          .ASCIZ  /BASIC KMC11 CONTROLLER TEST/<200>
(2)

        177570                  DSWR    =       177570
        177570                  DDISP   =       177570
```

```
                                .SBTTL  COMMON TAGS

                                ;;*****************************************************************
                                ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                ;*USED IN THE PROGRAM.

                        001200          .=1200
001200                  SCNTAG:                 ;;START OF COMMON TAGS
001200  000000                  .WORD   0
001202  000                     $ITNUM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001203  000                     $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001204  000000                  $SCNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001206  000000                  $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001210  000000                  $ERTTL: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001212  000000                  $ERMAX: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001214  000                     $ITCBY: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001215  001                     $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001216  000000                  $ERPC:  .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001220  000000                  $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001222  000000                  $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001224  000000                  $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001226  000000                  $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001230  000000                          .WORD   0               ;;RESERVED--NOT TO BE USED
001232  000000                          .WORD   0
001234  000                     $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001235  000                     $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001236  000000                          .WORD   0
001240  177570                  $SWR:   .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001242  177570                  DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
                                STKS=   177560                  ;;TTY KBD STATUS
                                STKB=   177562                  ;;TTY KBD BUFFER
                                STPS=   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
                                STPB=   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001244  000                     $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001245  002                     $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001246  012                     $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001247  000                     $TPFLG: .BYTE   0               ;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001250  000000                  $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                                ;WHICH  ($REG0) WAS OBTAINED
001252  000000                  $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001254  000000                  $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001256  000000                  $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001260  000000                  $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001262  000000                  $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001264  000000                  $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001266  000000                  $TMP0:  .WORD   0               ;;USER DEFINED
001270  000000                  $TMP1:  .WORD   0               ;;USER DEFINED
001272  000000                  $TMP2:  .WORD   0               ;;USER DEFINED
001274  000000                  $TMP3:  .WORD   0               ;;USER DEFINED
001276  000000                  $TMP4:  .WORD   0               ;;USER DEFINED
001300  000000                  $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001302  077                     $QUES:  .ASCII  /?/             ;;QUESTION MARK
001303  015                     $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001304  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE
```

```
                                ;;*****************************************************************
                                .EVEN
001316                          $MAIL:                  ;;APT MAILBOX
001316  000000                  $MSGTY: .WORD   AMSGTY  ;;MESSAGE TYPE CODE
001320  000000                  $FATAL: .WORD   AFATAL  ;;FATAL ERROR NUMBER
001322  000000                  $TESTN: .WORD   ATESTN  ;;TEST NUMBER
001324  000000                  $PASS:  .WORD   APASS       PASS COUNT
001326  000000                  $DEVCT: .WORD   ADEV        DEVICE COUNT
001330  000000                  $UNIT:  .WORD   AUNIT       UNIT NUMBER
001332  000000                  $MSGAD:                         ADDRESS
001334  000000                  $MSGLG:                         LENGTH
001336                          $ETABLE                     IRONMENT TABLE
001336      002                 $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
001337      000                 $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
001340  000000                  $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
001342  000000                  $USWR:  .WORD   AUSWR   ;;USER SWITCHES
001344  000000                  $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
                                ;*                      BITS 15-11=CPU TYPE
                                ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                ;*                              11/70=06,PDP=07,0=10
                                ;*                      BIT 10=REAL TIME CLOCK
                                ;*                      BIT  9=FLOATING POINT PROCESSOR
                                ;*                      BIT  8=MEMORY MANAGEMENT
001346      000                 $HAMS1: .BYTE   AHAMS1  ;;HIGH ADDRESS M.S. BYTE
001347      000                 $HTYP1: .BYTE   AHTYP1  ;;MEM. TYPE,BLK#1
                                ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
                                ;*                              000 NSEC CORE=001
                                ;*                              200 NSEC BIPOLAR=002
                                ;*                              400 NSEC MOS=003
001350  000000                  $HADR1: .WORD   AHADR1  ;;HIGH ADDRESS BLK#1
                                ;*                      MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
001352      000                 $HAMS2: .BYTE   AHAMS2  ;;HIGH ADDRESS M.S. BYTE
001353      000                 $HTYP2: .BYTE   AHTYP2  ;;MEM.TYPE,BLK#2
001354  000000                  $HADR2: .WORD   AHADR2  ;;MEM.LAST ADDRESS,BLK#2
001356      000                 $HAMS3: .BYTE   AHAMS3  ;;HIGH ADDRESS M.S.BYTE
001357      000                 $HTYP3: .BYTE   AHTYP3  ;;MEM.TYPE,BLK#3
001360  000000                  $HADR3: .WORD   AHADR3  ;;MEM.LAST ADDRESS,BLK#3
001362      000                 $HAMS4: .BYTE   AHAMS4  ;;HIGH ADDRESS M.S.BYTE
001363      000                 $HTYP4: .BYTE   AHTYP4  ;;MEM.TYPE,BLK#4
001364  000000                  $HADR4: .WORD   AHADR4  ;;MEM.LAST ADDRESS,BLK#4
001366  000000                  $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001370  000000                  $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001372  000000                  $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001374  000000                  $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
001376  000000                  $CON1:  .WORD   ACON1   ;;CONTROLLER DESCRIPTION WORD#1
001400  000000                  $CON2:  .WORD   ACON2   ;;CONTROLLER DESCRIPTION WORD#2
001402  000000                  $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
001404  000000                  $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
001406  000000                  $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
001410  000000                  $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
001412  000000                  $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
001414  000000                  $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
001416  000000                  $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
001420  000000                  $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
001422  000000                  $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
```

```
352  001424  000000                SDDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
353  001426  000000                SDDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
354  001430  000000                SDDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
355  001432  000000                SDDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
356  001434  000000                SDDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
357  001436  000000                SDDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
358  001440  000000                SDDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
359
360
361  001442                        SETEND:
362
363
364                                ;
                                   ;  PROGRAM CONTROL PARAMETERS
365                                ;---------------------------
366  001442  000000                NEXT:   .WORD   0               ; ADDRSS OF NEXT TEST TO BE EXECUTED
367  001444  000000                LOCK:   .WORD   0               ; ADDRESS FOR LOCK CURRENT DATA
368                                ;
369                                ;   PROGRAM VARIABLES
370                                ;-----------------------
371  001446  000000                STRTSW: .WORD   0               ; SWITCHES AT START OF PROGRAM
372  001450  000000                STAT:   .WORD   0               ; KM STATUS WORD STORAGE
373  001452  000000                CLKX:   .WORD   0               ;
374  001454  000000                MASKX:  .WORD   0               ;
375  001456  000000                SAVSP:  .WORD   0               ; STACK POINTER STORAGE
376  001460  000000                SAVPC:  .WORD   0               ; PROGRAM COUNTER STORAGE
377  001462  000000                ZERO:   .WORD   0               ;
378  001464  000001                ONE:    .WORD   1               ;
379  001466  000000                MEMLIM: .WORD   0               ; HIGHEST LOCATION FOR MPR'S
380  001470  000001                KMACTV: .BLKW   1               ; KMC11 SELECTED ACTIVE
381  001472  000001                KMNUM:  .BLKW   1               ; OCTAL NUMBER OF KMC11'S
382  001474  000001                SAVACT: .BLKW   1               ; ORIGINAL ACTIVE DEVICES.
383  001476  000001                SAVNUM: .BLKW   1               ; WORKABLE NUMBER.
384  001500  000000                RUN:    .WORD   0               ; POINTER TO RUNNING DEVICES
385                                        .EVEN
386  001502  002072                CREAM:  .WORD   KM.MAP-6        ; TABLE POINTER
387  001504  002276                MILK:   .WORD   CNT.MAP-4       ; TABLE POINTER
388                                ;
389                                ;  PROGRAM CONTROL FLAGS
390                                ;-----------------------
391  001506     000                INIFLG: .BYTE   0               ; PROGRAM INITIALIZING FLAG
392             001510                     .EVEN
393  001510     000                LOKFLG: .BYTE   0               ; LOCK ON CURRENT TEST FLAG
394  001511     000                QV.FLG: .BYTE   0               ; QUICK VERIFY FLAG
395                                                                ; ON FIRST PASS OF EACH KMC11 ITERATIONS WILL BE SUPPRES
396                                        .EVEN
```

```
                                .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 (SERRPC).
                                ;*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

001512                          SERRTB:
                                .EVEN
                                ;*      DF              ;; DOES NOT APPLY IN THIS DIAGNOSTIC.
001512  000000                          0
001514  000000                          0
001516  000000                          0
001520  036176                          EM1
001522  037256                          DH1             ; ERROR 1
001524  037434                          DT1
001526  036235                          EM2
001530  037256                          DH1             ; ERROR 2
001532  037434                          DT1
001534  036307                          EM3
001536  037317                          DH2             ; ERROR 3
001540  037456                          DT2
001542  036336                          EM4
001544  037256                          DH1             ; ERROR 4
001546  037434                          DT1
001550  036365                          EM5
001552  037256                          DH1             ; ERROR 5
001554  037434                          DT1
001556  036427                          EM6
001560  037334                          DH3             ; ERROR 6
001562  037434                          DT1
001564  036456                          EM7
001566  037334                          DH3             ; ERROR 7
001570  037434                          DT1
001572  036520                          EM10
001574  037334                          DH3             ; ERROR 10
001576  037434                          DT1
001600  036553                          EM11
001602  037317                          DH2             ; ERROR 11
001604  037456                          DT2
001606  036565                          EM12
001610  037317                          DH2             ; ERROR 12
001612  037456                          DT2
001614  036610                          EM13
001616  000000                          0               ; ERROR 13
001620  000000                          0
001622  036642                          EM14
001624  000000                          0               ; ERROR 14
001626  000000                          0
```

```
001630  036666                  EM15
001632  037373                  DH4             ; ERROR 15
001634  037434                  DT1
001636  036732                  EM16
001640  000000                  0               ; ERROR 16
001642  000000                  0
001644  037000                  EM17
001646  000000                  0               ; ERROR 17
001650  000000                  0
001652  037030                  EM20
001654  037317                  DH2             ; ERROR  20
001656  037456                  DT2
001660  037076                  EM21
001662  000000                  0               ; ERROR 21
001664  000000                  0
001666  037126                  EM22
001670  000000                  0               ; ERROR 22
001672  000000                  0
001674  037146                  EM23
001676  037373                  DH4             ; ERROR 23
001700  037434                  DT1
001702  037210                  EM24
001704  000000                  0               ; ERROR 24
001706  000000                  0
001710  037234                  EM25
001712  037317                  DH2             ; ERROR 25
001714  037456                  DT2
        002034          .=2034
                        .SBTTL  APT PARAMETER BLOCK

                        ;;**********************************************************************
                        ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                        ;;**********************************************************************
                                .$X=.   ;;SAVE CURRENT LOCATION
                                .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
                                200     ;;FOR APT START UP
                                .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
                                SAPTHDR ;;POINT TO APT HEADER BLOCK
                                .=.$X   ;;RESET LOCATION COUNTER
                        ;;**********************************************************************
                        ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                        ;INTERFACE SPEC.

002034                  SAPTHD:
002034  000000          SHIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
002036  001316          SMBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
002040  000132          STSTM:  .WORD   90.     ;;RUN TIM OF LONGEST TEST
002042  000137          SPASTM: .WORD   95.     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
002044  000137          SUNITM: .WORD   95.     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
002046  000052                  .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                                ;KMC11 CONTROL INDICATORS FOR CURRENT KMC11 UNDER TEST
                                ;-----------------------------------------------------

002050  000000                  STAT1:  0
002052  000000                  STAT2:  0
002054  000000                  STAT3:  0

                                ;KMC11 VECTOR AND REGISTER INDIRECT POINTERS
                                ;-------------------------------------------

002056  000000                  KMRVEC: 0               ;POINTER TO KMC11 RECEIVER INTERRUPT VECTOR
002060  000000                  KMRLVL: 0               ;POINTER TO KMC11 RECEIVER INTERRUPT SERVICE PS
002062  000000                  KMTVEC: 0               ;POINTER TO KMC11 TRANSMITTER INTERRUPT VECTOR
002064  000000                  KMTLVL: 0               ;POINTER TO KMC11 TRANSMITTER INTERRUPT SERVICE PS
002066  000000                  KMCSR:  0               ;POINTER TO KMC11 CONTROL STATUS REGISTER
002070  000000                  KMCSRH: 0               ;POINTER TO KMC11 CONTROL STATUS REGISTER HIGH BYTE.
002072  000000                  KMCTL:  0               ;POINTER TO KMC11 CONTOL OUT REGISTER
002074  000000                  KMP04:  0               ;POINTER TO KMC11 PORT REGISTER(SEL 4)
002076  000000                  KMP06:  0               ;POINTER TO KMC11 PORT REGISTER(SEL 6)

                                ;TEMP STORAGE
                                ;------------

                                ;TEMP:  0
                                ;.=.+40

                                ;KMC11 STATUS TABLE AND ADDRESS ASSIGNMENTS
                                ;------------------------------------------

                    002100      .=2100
002100                          KM.MAP:
002100  000001      KMCR00: .BLKW   1               ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 00
002102  000001      KMS100: .BLKW   1               ;VECTOR FOR KMC11 NUMBER 00
002104  000001      KMS200: .BLKW   1               ;DDCMP LINE# FOR KMC11 NUMBER 00
002106  000001      KMS300: .BLKW   1               ;3RD STATUS WORD

002110  000001      KMCR01: .BLKW   1               ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 01
002112  000001      KMS101: .BLKW   1               ;VECTOR FOR KMC11 NUMBER 01
002114  000001      KMS201: .BLKW   1               ;DDCMP LINE# FOR KMC11 NUMBER 01
002116  000001      KMS301: .BLKW   1               ;3RD STATUS WORD

002120  000001      KMCR02: .BLKW   1               ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 02
002122  000001      KMS102: .BLKW   1               ;VECTOR FOR KMC11 NUMBER 02
002124  000001      KMS202: .BLKW   1               ;DDCMP LINE# FOR KMC11 NUMBER 02
002126  000001      KMS302: .BLKW   1               ;3RD STATUS WORD

002130  000001      KMCR03: .BLKW   1               ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 03
002132  000001      KMS103: .BLKW   1               ;VECTOR FOR KMC11 NUMBER 03
002134  000001      KMS203: .BLKW   1               ;DDCMP LINE# FOR KMC11 NUMBER 03
002136  000001      KMS303: .BLKW   1               ;3RD STATUS WORD

002140  000001      KMCR04: .BLKW   1               ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 04
002142  000001      KMS104: .BLKW   1               ;VECTOR FOR KMC11 NUMBER 04
002144  000001      KMS204: .BLKW   1               ;DDCMP LINE# FOR KMC11 NUMBER 04
002146  000001      KMS304: .BLKW   1               ;3RD STATUS WORD
```

```
                                
002150  000001          KMCR05: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 05
002152  000001          KMS105: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 05
002154  000001          KMS205: .BLKW   1       ;DOCMP LINE# FOR KMC11 NUMBER 05
002156  000001          KMS305: .BLKW   1       ;3RD STATUS WORD
                                
002160  000001          KMCR06: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 06
002162  000001          KMS106: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 06
002164  000001          KMS206: .BLKW   1       ;DOCMP LINE# FOR KMC11 NUMBER 06
002166  000001          KMS306: .BLKW   1       ;3RD STATUS WORD
                                
002170  000001          KMCR07: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 07
002172  000001          KMS107: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 07
002174  000001          KMS207: .BLKW   1       ;DOCMP LINE# FOR KMC11 NUMBER 07
002176  000001          KMS307: .BLKW   1       ;3RD STATUS WORD
                                
002200  000001          KMCR10: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 10
002202  000001          KMS110: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 10
002204  000001          KMS210: .BLKW   1       ;DOCMP LINE# FOR KMC11 NUMBER 10
002206  000001          KMS310: .BLKW   1       ;3RD STATUS WORD
                                
002210  000001          KMCR11: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 11
002212  000001          KMS111: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 11
002214  000001          KMS211: .BLKW   1       ;DOCMP LINE# FOR KMC11 NUMBER 11
002216  000001          KMS311: .BLKW   1       ;3RD STATUS WORD
                                
002220  000001          KMCR12: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 12
002222  000001          KMS112: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 12
002224  000001          KMS212: .BLKW   1       ;DOCMP LINE# FOR KMC11 NUMBER 12
002226  000001          KMS312: .BLKW   1       ;3RD STATUS WORD
                                
002230  000001          KMCR13: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 13
002232  000001          KMS113: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 13
002234  000001          KMS213: .BLKW   1       ;DOCMP LINE# FOR KMC11 NUMBER 13
002236  000001          KMS313: .BLKW   1       ;3RD STATUS WORD
                                
002240  000001          KMCR14: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 14
002242  000001          KMS114: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 14
002244  000001          KMS214: .BLKW   1       ;DOCMP LINE# FOR KMC11 NUMBER 14
002246  000001          KMS314: .BLKW   1       ;3RD STATUS WORD
                                
002250  000001          KMCR15: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 15
002252  000001          KMS115: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 15
002254  000001          KMS215: .BLKW   1       ;DOCMP LINE# FOR KMC11 NUMBER 15
002256  000001          KMS315: .BLKW   1       ;3RD STATUS WORD
                                
002260  000001          KMCR16: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 16
002262  000001          KMS116: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 16
002264  000001          KMS216: .BLKW   1       ;DOCMP LINE# FOR KMC11 NUMBER 16
002266  000001          KMS316: .BLKW   1       ;3RD STATUS WORD
                                
002270  000001          KMCR17: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 17
002272  000001          KMS117: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 17
002274  000001          KMS217: .BLKW   1       ;DOCMP LINE# FOR KMC11 NUMBER 17
002276  000001          KMS317: .BLKW   1       ;3RD STATUS WORD
```

L03


```
   616  002300  000000                KM.END: 000000
```

```
                        ;KMC11 PASS COUNT AND ERROR COUNT TABLE
                        ;--------------------------------------

002302                  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

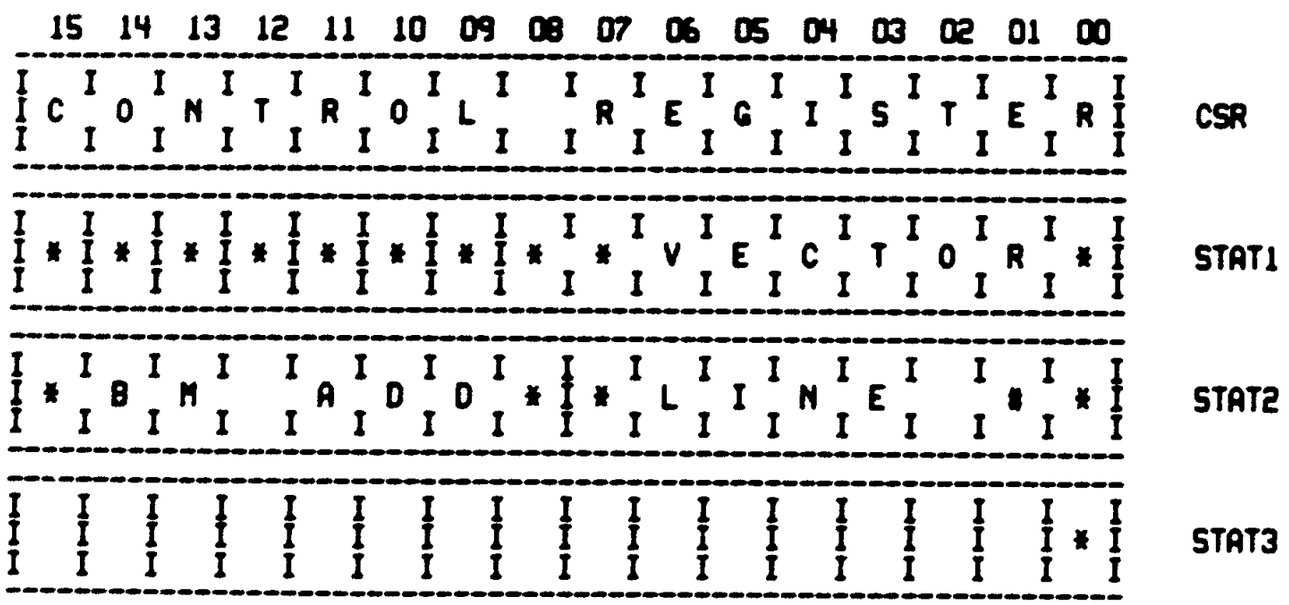
002366  000000          PACT15: 0               ;PASS COUNT FOR KMC11 NUMBER 15
002370  000000          ERCT15: 0               ;ERROR COUNT FOR KMC11 NUMBER 15

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



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 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: BIT0=1 DO FREE RUNNING TESTS ON KMC
(MUST BE SET TO A ONE MANUALLY [PROGRAMS G AND H ONLY])

```
                                        ;PROGRAM INITIALIZATION
                                        ;LOCK OUT INTERRUPTS
                                        ;SET UP PROCESSOR STACK
                                        ;SET UP POWER FAIL VECTOR
                                        ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
                                        ;TYPE TITLE MESSAGE

002402  012737  000340  177776  .START: MOV     #340,PS         ;LOCK OUT INTERRUPTS
002410  012706  001200          MOV     #STACK,SP       ;SET UP STACK
002414  012737  007100  000024  MOV     #SPWRDN,@#24    ;SET UP POWER FAIL VECTOR
002422  013737  001472  001476  MOV     KNNUM,SAVNUM    ;SAVE NUMBER OF DEVICES IN SYSTEM.
002430  005037  011422          CLR     SWFLG           ;CLEAR SOFT TYPEOUT FLAG
002434  105037  001203          CLRB    ERRFLG          ;CLEAR ERROR FLAG
002440  105037  001511          CLRB    QV.FLG          ;ZERO QUICK VERIFY FLAG
002444  012737  002070  001502  MOV     #KM.MAP-10,CREAM;GET MAP POINTER.
002452  012737  002276  001504  MOV     #CNT.MAP-4,MILK ;GET PASS COUNT MAP POINTER
002460  012737  100000  001500  MOV     #BIT15,RUN      ;POINT POINTER TO FIRST DEVICE.
002466  012700  002302          MOV     #CNT.MAP,R0     ;PASS COUNT POINTER TO R0
002472  005020          23$:    CLR     (R0)+           ;CLEAR TABLE
002474  022700  002402          CMP     #CNT.MAP+100,R0 ;DONE YET?
002500  001374          BNE     23$             ;KEEP GOING
002502  005037  001216          CLR     SERRPC          ;CLEAR LAST ERROR POINTER
002506  012737  000001  0       MOV     #1,$TSTNM       ;SET UP FOR TEST 1
002514  012737  002402  0       MOV     #.START,SLPADR  ;SET UP FOR POWER FAIL BEFORE
                                                        ;TESTING STARTS
002522  132737  000001          BITB    #1,$ENV         ; IS IT RUNNING UNDER APT?
002530  001404          BEQ     3$              ; IF NOT CHECK FOR TYPE OF SWITCH REGISTER.
002532  013737  001340          MOV     $SWREG,SWREG    ; LOAD SOFTWARE SWITCH REG.
002540  000423          BR      6$+2            ; GO SET UP SOFTWARE SWITCH REG.
002542  013746  000006          3$:     MOV     @#6,-(SP)       ;SAVE CURRENT VECTORS
002546  013746  000004          MOV     @#4,-(SP)
002552  012737  002606  000004  MOV     #6$,@#4         ;SET UP FOR TIMEOUT
002560  012737  177570  001240  MOV     #177570,SWR     ;SET SWR TO HARD SWR ADDRESS
002566  012737  177570  001242  MOV     #177570,DISPLAY ;SET DISPLAY TO HARD SWR ADDRESS
002574  022777  177777  176436  CMP     #-1,@SWR        ;REFERENCE HARDWARE SWITCH REGISTER
002602  001402          BEQ     6$+2            ;IF = -1 USE SOFT SWR ANYWAY
002604  000407          BR      7$              ;IF IT EXISTS AND NOT = -1 USE HARD SWR
002606  022626          6$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
002610  012737  000176  001240  MOV     #SWREG,SWR      ;POINTER TO SOFT SWR
002616  012737  000174  001242  MOV     #DISPREG,DISPLAY;POINTER TO SOFT DISPLAY REG
002624  012637  000004          7$:     MOV     (SP)+,@#4       ;RESTORE VECTORS
002630  012637  000006          MOV     (SP)+,@#6
002634  105737  001506          TSTB    INIFLG          ;HAS INITIALIZATION BEEN PERFORMED
002640  001006          BNE     20$             ;BR IF YES
002642  022737  004070  000042  CMP     #SEN040,@#42    ;IF ACT-11 AUTOMATIC MODE, DON'T TYPE ID
002650  001402          BEQ     20$
002652  104401  001000          TYPE    ,MTITLE         ;TYPE TITLE MESSAGE
002656  004737  011216          20$:    JSR     PC,CKSWR                ;CHECK FOR SOFT SWR
002662  017737  176352  001446  MOV     @SWR,STRTSW     ;STORE STARTING SWITCHES
002670  005737  000042          TST     @#42            ;IS IT RUNNING IN AUTO MODE?
002674  001402          BEQ     .+6             ;BR IF NO
002676  005037  001446          CLR     STRTSW          ;IF YES, CLEAR SWITCHES
002702  032737  000001  001446  BIT     #SW00,STRTSW    ;IF SW00=1, QUESTIONS ARE ASKED.
002710  001012          BNE     17$             ;BR IF SW00=1
002712  105737  001446          TSTB    STRTSW          ;BIT7=1??
```

```
 780  002716  100007                        BPL     17$             ;BR IF SWD7=0
 781  002720  005737  001470                TST     KMACTV          ;ARE ANY DEVICES SELECTED?
 782  002724  001027                        BNE     16$             ;BR IF YES
 783  002726  104401  010703                TYPE,   NOACT           ;NO DEVICES SELECTED.
 784  002732  000000                        HALT                    ;STOP THE SHOW
 785  002734  000776                        BR      .-2             ;DISQUALIFY CONTINUE SWITCH
 786  002736  105737  001336        17$:    TSTB    $ENV            ;  IS IT UNDER APT DUMP MODE?
 787  002742  001405                        BEQ     27$             ;  YES, CHECK IF APT SIZED IT?
 788  002744  032737  000001  001336        BITB    #1,$ENV         ;  IS IT UNDER D,V OR RUN MODE?
 789  002752  001012                        BNE     30$             ;  YES, NEEDS ONLY APT SIZING.
 790  002754  000406                        BR      33$             ;  NO, NEEDS REGULAR AUTO.SIZE.
 791  002756  105737  001337        27$:    TSTB    $ENVM           ;  IS IT SIZED BY APT?
 792  002762  100406                        BMI     30$             ;  YES, NEEDS ONLY APT SIZING.
 793  002764  042737  000001  001446        BIC     #SWDD,STRTSW    ;  SIZE ONLY IN AUTO MODE.
 794  002772  004737  012152        33$:    JSR     PC,AUTO.SIZE    ;  GO DO THE AUTO.SIZE.
 795  002776  000402                        BR      16$             ;  GO PRINT THE MAP.
 796  003000  004737  013552        30$:    JSR     PC,APT.SIZE     ;  GO DO THE APT SIZING.
 797  003004  105737  001506        16$:    TSTB    INIFLG          ;FIRST TIME?
 798  003010  001410                        BEQ     21$             ;BR IF YES
 799  003012  105737  001446                TSTB    STRTSW          ;IF USING SAME PARAMETERS DONT TYPE MAP
 800  003016  100431                        BMI     1$
 801  003020  032737  000006  001446        BIT     #BIT1!BIT2,STRTSW;IS TEST NO. OR LOCK SELECTED
 802  003026  001403                        BEQ     24$             ;IF NO THEN TYPE STATUS
 803  003030  000424                        BR      1$              ;IF YES DO NOT TYPE STATUS
 804  003032  105137  001506        21$:    COMB    INIFLG          ;SET FLAG
 805  003036  104401  010051        24$:    TYPE    ,XHEAD          ;TYPE HEADER
 806  003042  012704  002100                MOV     #KM.MAP,R4      ;SET POINTER
 807  003046  010437  001276        5$:     MOV     R4,STMP0        ;SET ADDRESS
 808  003052  012437  001300                MOV     (R4)+,STMP1     ;SET CSR
 809  003056  001411                        BEQ     1$              ;ALL DONE IF ZERO
 810  003060  012437  001302                MOV     (R4)+,STMP2     ;SET STAT1
 811  003064  012437  001304                MOV     (R4)+,STMP3     ;SET STAT2
 812  003070  012437  001306                MOV     (R4)+,STMP4     ;SET STAT3
 813  003074  104416                        CONVRT                  ;TYPE OUT STATUS MAP
 814  003076  011064                        XSTATQ                  ;
 815  003100  000762                        BR      5$
 816  003102  012700  002100        1$:     MOV     #KM.MAP,R0      ;R0 POINTS TO STATUS TABLE
 817
 818                                ;;*****************************************************************
 819                                ;;*AUTO SIZE TEST
 820                                ;;*THIS TEST VERIFYS THAT THE KMC11S AND/OR KMC11S ARE AT THE CORRECT FLOATING
 821                                ;;*ADDRESSES FOR YOUR SYSTEM. IF THIS TEST FAILS, IT IS NOT A HARDWARE ERROR.
 822                                ;;*CHECK THE ADDRESSES OF ALL FLOATING DEVICES (DJ,DH,DQ,DU,DUP,LK,DMC,DZ,KMC).
 823                                ;;*IF THERE ARE NO OTHER FLOATING DEVICES BEFORE THE KMC11, THE FIRST
 824                                ;;*KMC11 IS 760110. NO DEVICE SHOULD EVER BE AT
 825                                ;;*ADDRESS 760000.
 826                                ;;*****************************************************************
 827
 828  003106  013746  000004                MOV     @#4,-(SP)       ;SAVE LOC 4
 829  003112  013746  000006                MOV     @#6,-(SP)       ;SAVE LOC 6
 830  003116  005037  000006                CLR     @#6             ;CLEAR VEC+2
 831  003122  005037  001302                CLR     STMP2           ;CLEAR FLAG
 832  003126  011037  002066        AUSTRT: MOV     (R0),KMCSR      ;GET NEXT KMC CSR
 833  003132  001510                        BEQ     AUDONE          ;BR IF DONE
 834  003134  012737  003240  000004 2$:     MOV     #NODEV,@#4      ;SET UP FOR TIMEOUT
 835  003142  012703  000010        3$:     MOV     #10,R3          ;R3 IS COUNT OF DEVICES BEFORE KMC
```

```
003146  012702  003342          4$:     MOV     #DEVTAB,R2      ;R2 IS DEVICE TABLE PONTER
003152  012701  160010                  MOV     #160010,R1      ;START WITH ADDRESS 160010
003156  005711                  FLOAT:  TST     (R1)            ;CHECK ADDRESS IN R1
003160  111204                          MOVB    (R2),R4         ;IF NO TIMEOUT, GET NEXT ADDRESS
003162  060401                          ADD     R4,R1           ;IN R1
003164  005201                          INC     R1
003166  040401                          BIC     R4,R1
003170  005703                          TST     R3              ;ANY MORE DEVICES TO CHECK FOR?
003172  001371                          BNE     FLOAT           ;BR IF YES
003174  012737  003244  000004          MOV     #ERR,@#4        ;OK ONLY KMC'S ARE LEFT, SET UP FOR TIMEOUT
003202  005711                  FY:     TST     (R1)            ;CHECK KMC ADDRESS
003204  020137  002066                  CMP     R1,KMCSR        ;DOES IT MATCH
003210  001403                          BEQ     OK              ;BR IF YES
003212  062701  000010                  ADD     #10,R1          ;GET NEXT KMC ADDRESS
003216  000771                          BR      FY              ;DO IT AGAIN
003220  062700  000010          OK:     ADD     #10,R0          ;SKIP TO NEXT KMC CSR
003224  062701  000010                  ADD     #10,R1          ; GET NEXT KMC ADDRESS
003230  011037  002066                  MOV     (R0),KMCSR      ; GET NEXT KMC CSR
003234  001447                          BEQ     AUDONE          ; BRANCH IF ALL DONE.
003236  000761                          BR      FY              ; DO IT AGAIN.
003240  122243                  NODEV:  CMPB    (R2)+,-(R3)     ;ON TIMEOUT, INC R2, DEC R3
003242  000002                          RTI                     ;SLPADR
003244  005737  001302          ERR:    TST     $TMP2           ;CHECK FLAG IF = 0 TYPE HEADER
003250  001014                          BNE     1$              ;SKIP HEADER
003252  104401                          TYPE                    ;TYPEOUT HEADER MESSAGE
003254  010734                          CONERR                  ;CONFIGURATION ERROR!!!!
003256  012737  003244  001460          MOV     #ERR,SAVPC      ;SAVE PC FOR TYPEOUT
003264  104417                          CNVRT                   ;TYPE OUT ERROR PC
003266  003322                          ERRPC                   ;
003270  104401                          TYPE                    ;TYPE REST OF HEADER
003272  011001                          CNERR                   ;
003274  012737  177777  001302          MOV     #-1,$TMP2       ;SET FLAG SO IT ONLY GETS TYPED ONCE
003302  010137  001264          1$:     MOV     R1,$REG1        ;SAVE R1 FOR TYPEOUT
003306  104416                          CONVRT
003310  003330                          CONTAB                  ;TYPE CSR VALUES
003312  104401                  3$:     TYPE
003314  011022                          KMCM
003316  022626                  4$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
003320  000737                          BR      OK              ;BR TO GET OUT
003322  000001                  ERRPC:  1
003324      006         002             .BYTE   6,2
003326  001460                          SAVPC
003330  000002                  CONTAB: 2
003332      006         004             .BYTE   6,4
003334  001264                          $REG1
003336      006         002             .BYTE   6,2
003340  002066                          KMCSR
003342      007                 DEVTAB: .BYTE   7               ;DJ
003343      017                         .BYTE   17              ;DH
003344      007                         .BYTE   7               ;DQ
003345      007                         .BYTE   7               ;DU
003346      007                         .BYTE   7               ;DUP
003347      007                         .BYTE   7               ;LK
003350      007                         .BYTE   7               ;DMC
003351      007                         .BYTE   7               ;DZ
003352      007                         .BYTE   7               ;KMC
```

PROGRAM INITIALIZATION AND START UP.

```
                    003354                  .EVEN
003354                                      AUDONE:
003354  012637  000006                1$:   MOV     (SP)+,@#6       ;RESTORE LOC 6
003360  012637  000004                      MOV     (SP)+,@#4       ;RESTORE LOC 4
003364  032737  000010  001446              BIT     #SW03,STRTSW    ;SELECT SPECIFIC DEVICES??
003372  001422                            BEQ     3$              ;BR IF NO.
003374  104401  007771                    TYPE    ,MNEW           ;TYPE THE MESSAGE.
003400  005000                            CLR     R0              ;ZERO DATA LIGHTS
003402  000000                            HALT                    ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
003404  027737  175630  001474            CMP     @SWR,SAVACT     ;IS THE NUMBER VALID?
003412  101404                            BLOS    2$              ;BR IF NUMBER IS OK.
003414  104401  007644                    TYPE    ,MERR3          ;TELL USER OF INVALID NUMBER.
003420  000000                            HALT                    ;STOP EVERY THING.
003422  000776                            BR      .-2             ;RESTART THE PROGRAM AGAIN.
003424  017737  175610  001470      2$:   MOV     @SWR,KMACTV     ;GET NEW DEVICE PATTERN
003432  013700  001470                    MOV     KMACTV,R0       ;SHOW THE USER WHAT HE SELECTED.
003436  000000                            HALT                    ;CONTINUE DYNAMIC SWITCHES.
003440  012700  000300              3$:   MOV     #300,R0         ;PREPARE TO CLEAR THE FLOATING
003444  012701  000302                    MOV     #302,R1         ;VECTOR AREA. 300-776
003450  010120                      4$:   MOV     R1,(R0)+        ;START PUTTING "PC+2 - HALT"
003452  005021                            CLR     (R1)+           ;IN VECTOR AREA.
003454  022021                            CMP     (R0)+,(R1)+     ;POP POINTERS
003456  022700  001000                    CMP     #1000,R0        ;ALL DONE??
003462  001372                            BNE     4$              ;BR IF NO.

                                        ;TEST START AND RESTART
                                        ;----------------------

003464  012706  001200            .BEGIN: MOV     #STACK,SP       ;SET UP STACK
003470  013746  000006                    MOV     @#6,-(SP)       ;SAVE LOC 6
003474  013746  000004                    MOV     @#4,-(SP)       ;SAVE LOC 4
003500  005000                            CLR     R0              ;START AT 0
003502  012737  003546  000004            MOV     #2$,@#4         ;SET UP FOR TIME OUT
003510  005037  000006                    CLR     @#6             ;TO AUTOSIZE MEMORY
003514  005720                      6$:   TST     (R0)+           ;CHECK ADDRESS IN R0
003516  022700  157776                    CMP     #157776,R0      ;IS IT AT LEAST 28K
003522  001374                            BNE     6$              ;BR IF NO
003524  162700  007776                    SUB     #7776,R0        ;SAVE 2K FOR MONITORS
003530  010037  001466              7$:   MOV     R0,MEMLIM       ;STORE MEMORY LIMIT
003534  012637  000004                    MOV     (SP)+,@#4       ;RESTORE LOC 4
003540  012637  000006                    MOV     (SP)+,@#6       ;RESTORE LOC 6
003544  000413                            BR      10$             ;CONTINUE
003546  022626                      2$:   CMP     (SP)+,(SP)+     ;ADJUST STACK
003550  162700  000004                    SUB     #4,R0           ;GET LAST GOOD ADDRESS
003554  162700  007776                    SUB     #7776,R0        ;SAVE 2K FOR MONITORS
003560  022700  030000                    CMP     #30000,R0       ;IS IT 8K?
003564  001361                            BNE     7$              ;BR IF NO
003566  012700  037400                    MOV     #37400,R0       ;IF 8K DON'T SAVE 2K
003572  000756                            BR      7$
003574  012737  000340  177776    10$:    MOV     #340,PS         ;LOCK OUT INTERRUPTS
003602  032737  000004  001446            BIT     #BIT2,STRTSW    ;CHECK FOR LOCK ON TEST
003610  001406                            BEQ     1$              ;BR IF NO LOCK DESIRED.
003612  104401  007670                    TYPE    ,MLOCK          ;TYPE LOCK SELECTED.
003616  012737  000240  004146            MOV     #NOP,TTST       ;SET UP TO LOCK
003624  000403                            BR      3$              ;CONTINUE ALONG.
003626  013737  004354  004146    1$:     MOV     BRW,TTST        ;PREPARE NORMAL SCOPE ROUTINE
```

```
948  003634  012737  011464  001206  3$:     MOV     #CYCLE,$LPADR   ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
949  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$:     JMP     @$LPADR         ;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 ($PASS)
                ;*IF THERES A MONITOR GO TO IT
                ;*IF THERE ISN'T JUMP TO CYCLE

003662          $EOP:
003662 000005           RESET
003664 005237 001324    INC     $PASS           ;INCREMENT THE PASS COUNT
003670 105037 001203    CLRB    $ERFLG          ;CLEAR ERROR FLAG
003674 104401 007572    TYPE    ,MEPASS         ;TYPE END PASS.
003700 104401 007717    TYPE    ,MCSRX          ;TYPE "CSR"
003704 104417 004104    CNVRT   ,XCSR           ;SHOW IT.
003710 104401 007725    TYPE    ,MVECX          ;TYPE VECTOR.
003714 104417 004112    CNVRT   ,XVEC           ;SHOW IT.
003720 104401 007733    TYPE    ,MPASSX         ;TYPE " PASSES -
003724 104417 004120    CNVRT   ,XPASS          ;SHOW IT.
003730 104401 007744    TYPE    ,MERRX          ;TYPE " ERRORS -
003734 104417 004126    CNVRT   ,XERR           ;SHOW IT.
003740 013700 001504    MOV     #ILK,R0         ;SET POINTER TO PASSCNT.
003744 013720 001324    MOV     $PASS,(R0)+     ;SAVE THE PASS COUNT.
003750 013720 001212    MOV     $ERTTL,(R0)+    ;SAVE ERROR COUNT
003754 013777 002060 176074     MOV     @KWRLVL,@#KWRVEC        ;RESTORE THE RECEIVER INTERRUPT VECTOR.
003762 005077 176072    CLR     @KWRLVL         ;RESTORE RECEIVER LEVEL
003766 013777 002064 176066     MOV     @KWTLVL,@#KWTVEC        ;RESTORE THE TRANSMIT INTERRUPT VECTOR.
003774 005077 176064    CLR     @KWTLVL         ;RESTORE TRANSMITTER LEVEL
004000 005337 001476    DEC     $DVNUM          ;ALL DEVICE TESTED?
004004 001035           BNE     $DOAGN          ;BRANCH IF NO.
004006 112737 000377 001511     MOVB    #377,GV.FLG     ;SET QUICK VERIFY FLAG.
004014 013737 001472 001476     MOV     $KWNUM,$DVNUM   ;RESTORE DEVICE COUNT.
004022 005037 001216    CLR     $ERRPC          ;CLEAR LAST ERROR PC
004026 005037 001310    CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
004032 005237 001324    INC     $PASS           ;;INCREMENT THE PASS NUMBER
004036 042737 100000 001324     BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
004044 005327           DEC     (PC)+           ;;LOOP?
004046 000001   $EOPCT: .WORD   1
004050 003013           BGT     $DOAGN          ;;YES
004052 012737           MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
004054 000001   $ENDCT: .WORD   1
004056 004046           $EOPCT
004060 013700 000042    $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
004064 001405           BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
004066 000005           RESET                   ;;CLEAR THE WORLD
004070 004710   $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
004072 000240           NOP                     ;;SAVE ROOM
004074 000240           NOP                     ;;FOR
004076 000240           NOP                     ;;ACT11
004100          $DOAGN:
004100 000137           JMP     @(PC)+          ;;RETURN
```

```
                        ;END OF PASS ROUTINE

:009  004102  011464              $RTNAD: .WORD   CYCLE
1010  004104  000001              XCSR:   1
1011  004106     006      002             .BYTE   6,2
1012  004110  002066              KMCSR
1013  004112  000001              XVEC:   1
1014  004114     004      002             .BYTE   4,2
1015  004116  002056              KMRVEC
1016  004120  000001              XPASS:  1
1017  004122     006      002             .BYTE   6,2
1018  004124  001324              $PASS
1019  004126  000001              XERR:   1
1020  004130     006      002             .BYTE   6,2
1021  004132  001212              $ERTTL
1022
1023                              ;SCOPE LOOP AND INTERATION HANDLER
1024                              ;-------------------------------
1025
1026                        .SBTTL  SCOPE HANDLER ROUTINE
1027
1028                        ;;**************************************************************
1029                        ;#THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
1030                        ;#AND LOAD THE TEST NUMBER($ISTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
1031                        ;#AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
1032                        ;#THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
1033                        ;#SW14=1         LOOP ON TEST
1034                        ;#SW11=1         INHIBIT ITERATIONS
1035                        ;#CALL
1036                        ;#      SCOPE            ;;SCOPE=IOT
1037
1038  004134                $SCOPE:
1039  004134  005037  001216          CLR     $ERRPC                  ; CLEAR LAST ERROR PC
1040  004140  023716  013776          CMP     TST1+2,(SP)             ; IS THIS TEST #1 ?
1041  004144  001413                  BEQ     $XTSTR                  ; IF SO DON'T LOOP.
1042  004146  000406          TTST:   BR      1$
1043  004150  105777  175070          TSTB    @$TKS                   ; KEYBOARD DONE ?
1044  004154  100067                  BPL     $OVER                   ; IF NO DONT WAIT.
1045  004156  017766  175064  177776        MOV     @$TKB,-2(SP)
1046  004164  032777  040000  175046  1$:     BIT     #BIT14,@$WR     ;;LOOP ON PRESENT TEST?
1047  004172  001060                  BNE     $OVER                   ;;YES IF SW14=1
1048                        ;#####START OF CODE FOR THE XOR TESTER#####
1049  004174  000416          $XTSTR: BR      6$                      ;;IF RUNNING ON THE "XOR" TESTER CHANGE
1050                                                                  ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
1051  004176  013746  000004          MOV     @#ERRVEC,-(SP)          ;;SAVE THE CONTENTS OF THE ERROR VECTOR
1052  004202  012737  004222  000004  MOV     #5$,@#ERRVEC            ;;SET FOR TIMEOUT
1053  004210  005737  177060          TST     @#177060                ;;TIME OUT ON XOR?
1054  004214  012637  000004          MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
1055  004220  000436                  BR      $SVLAD                  ;;GO TO THE NEXT TEST
1056  004222  022626          5$:     CMP     (SP)+,(SP)+             ;;CLEAR THE STACK AFTER A TIME OUT
1057  004224  012637  000004          MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
1058  004230  000441                  BR      $OVER                   ;;LOOP ON THE PRESENT TEST
1059  004232                  6$:;#####END OF CODE FOR THE XOR TESTER#####
1060  004232  105737  001203          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
1061  004236  001404                  BEQ     3$                      ;;BR IF NO
1062  004240  105037  001203          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
1063  004244  005037  001310                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
1064  004250  032777  004000  174762  3$:     BIT     #BIT11,@$WR     ;;INHIBIT ITERATIONS?
```

```
1065  004256  001011                    BNE     1$              ;;BR IF YES
1066  004260  005737  001324            TST     $PASS           ;;IF FIRST PASS OF PROGRAM
1067  004264  001406                    BEQ     1$              ;;       INHIBIT ITERATIONS
1068  004266  005237  001204            INC     $ICNT           ;;INCREMENT ITERATION COUNT
1069  004272  023737  001310  001204    CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
1070  004300  002015                    BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
1071  004302  012737  000001  001204  1$:       MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
1072  004310  013737  004356  001310            MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
1073  004316  105237  001202        $SVLAD:     INCB    $TSTNM          ;;COUNT TEST NUMBERS
1074  004322  113737  001202  001322            MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
1075  004330  011637  001206                    MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
1076  004334  013777  001202  174700  $OVER:    MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
1077  004342  013716  001206                    MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
1078  004346  013701  002066                    MOV     KMCSR,R1                ; R1 CONTAINS BASE KMC ADDRESS.
1079  004352  000002                    RTI
1080  004354  000406            BRW:    .WORD   406
1081  004356  000020            $MXCNT: 20                      ;;MAX. NUMBER OF ITERATIONS
1082
1083                                    ;CHECK FOR FREEZE ON CURRENT DATA
1084                                    ;--------------------------------
1085
1086  004360  004737  011216        .SCOP1: JSR     PC,CKSWR                ;CHECK FOR SOFT SWR
1087  004364  032777  001000  174646        BIT     #SW09,@SWR      ;IS SW09=1(SET)?
1088  004372  001403                    BEQ     1$              ;BR IF NOT SET.
1089  004374  012737  177777  001444        MOV     #-1,LOCK        ;FLAG IT.
1090  004402  000002            1$:     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:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
1101                                ;#NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
1102                                ;#NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
1103                                ;*
1104                                ;#CALL:
1105                                ;#1) USING A TRAP INSTRUCTION
1106                                ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
1107                                ;#OR
1108                                ;*      TYPE
1109                                ;*      MESADR
1110                                ;*
1111
1112  004404  105737  001257        STYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
1113  004410  100002                        BPL     1$              ;;BR IF YES
1114  004412  000000                        HALT                    ;;HALT HERE IF NO TERMINAL
1115  004414  000430                        BR      3$              ;;LEAVE
1116  004416  010046                1$:     MOV     R0,-(SP)        ;;SAVE R0
1117  004420  017600  000002                MOV     2(SP),R0        ;;GET ADDRESS OF ASCIZ STRING
1118  004424  122737  000001  001336        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
1119  004432  001011                        BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
1120  004434  132737  000100  001337        BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
```

```
1121  004442  001405                    BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
1122  004444  010037  004454            MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
1123  004450  004737  004674            JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
1124  004454  000000              61$:    .WORD   0               ;;MESSAGE ADDRESS
1125  004456  132737  000040  001337    BITB    #APTCSUP,SENVM  ;;APT CONSOLE SUPPRESSED
1126  004464  001003                    BNE     62$             ;;YES,SKIP TYPE OUT
1127  004466  112046              2$:   MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
1128  004470  001005                    BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
1129  004472  005726                    TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
1130  004474  012600              62$:  MOV     (SP)+,R0        ;;RESTORE R0
1131  004476  062716  000002      3$:   ADD     #2,(SP)         ;;ADJUST RETURN PC
1132  004502  000002                    RTI                     ;;RETURN
1133  004504  122716  000011      4$:   CMPB    #HT,(SP)        ;;BRANCH IF <HT>
1134  004510  001430                    BEQ     8$
1135  004512  122716  000200            CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
1136  004516  001006                    BNE     5$
1137  004520  005726                    TST     (SP)+           ;;POP  <CR><LF> EQUIV
1138  004522  104401                    TYPE                    ;;TYPE A CR AND LF
1139  004524  001315                    $CRLF
1140  004526  105037  004662            CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
1141  004532  000755                    BR      2$              ;;GET NEXT CHARACTER
1142  004534  004737  004616      5$:   JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
1143  004540  123726  001256      6$:   CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
1144  004544  001350                    BNE     2$              ;;IF NO GO GET NEXT CHAR.
1145  004546  013746  001254            MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
1146                                                            ;;AND THE NULL CHAR.
1147  004552  105366  000001      7$:   DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
1148  004556  002770                    BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
1149  004560  004737  004616            JSR     PC,$TYPEC       ;;GO TYPE A NULL
1150  004564  105337  004662            DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
1151  004570  000770                    BR      7$              ;;LOOP
1152
1153                                    ;HORIZONTAL TAB PROCESSOR
1154
1155  004572  112716  000040      8$:   MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
1156  004576  004737  004616      9$:   JSR     PC,$TYPEC       ;;TYPE A SPACE
1157  004602  132737  000007  004662    BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
1158  004610  001372                    BNE     9$              ;;TAB STOP
1159  004612  005726                    TST     (SP)+           ;;POP SPACE OFF STACK
1160  004614  000724                    BR      2$              ;;GET NEXT CHARACTER
1161  004616  105777  174426      $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
1162  004622  100375                    BPL     $TYPEC
1163  004624  116677  000002  174420    MOVB    2(SP),@$TPB     ;;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     1$              ;;BRANCH IF NO
1166  004642  105037  004662            CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
1167  004646  000406                    BR      $TYPEX          ;;EXIT
1168  004650  122766  000012  000002  1$:   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      $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
1172  004664  000207      $TYPEX: RTS     PC
1173
1174                                    .SBTTL  APT COMMUNICATIONS ROUTINE
1175
1176                                    ;;**************************************************************************
```

```
1177  004666  112737  000001  005132  $ATY1:  MOVB    #1,$EFLG        ;;TO REPORT FATAL ERROR
1178  004674  112737  000001  005130  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
1179  004702  000403                  BR      $ATYC
1180  004704  112737  000001  005132  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
1181  004712                          $ATYC:
1182  004712  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
1183  004714  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
1184  004716  105737  005130          TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
1185  004722  001450                  BEQ     5$              ;;IF NOT:  BR
1186  004724  122737  000001  001336  CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
1187  004732  001031                  BNE     3$              ;;IF NOT:  BR
1188  004734  132737  000100  001337  BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
1189  004742  001425                  BEQ     3$              ;;IF NOT:  BR
1190  004744  017600  000004          MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
1191  004750  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
1192  004756  005737  001316  1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
1193  004762  001375                  BNE     1$              ;;IF NOT:  WAIT
1194  004764  010037  001332          MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
1195  004770  105720          2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
1196  004772  001376                  BNE     2$
1197  004774  163700  001332          SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
1198  005000  006200                  ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
1199  005002  010037  001334          MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
1200  005006  012737  000004  001316  MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
1201  005014  000413                  BR      5$
1202  005016  017637  000004  005042  3$:     MOV     @4(SP),4$       ;;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,$TYPE        ;;CALL TYPE MACRO
1206  005042  000000          4$:     .WORD   0
1207  005044                  5$:
1208  005044  105737  005132  10$:     TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
1209  005050  001416                  BEQ     12$             ;;IF NOT:  BR
1210  005052  005737  001336          TST     $ENV            ;;RUNNING UNDER APT?
1211  005056  001413                  BEQ     12$             ;;IF NOT:  BR
1212  005060  005737  001316  11$:     TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
1213  005064  001375                  BNE     11$             ;;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     $MSGTYPE        ;;TELL APT TO TAKE ERROR
1217  005106  105037  005132  12$:     CLRB    $FFLG           ;;CLEAR FATAL FLAG
1218  005112  105037  005131          CLRB    $LFLG           ;;CLEAR LOG FLAG
1219  005116  105037  005130          CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
1220  005122  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
1221  005124  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
1222  005126  000207                  RTS     PC              ;;RETURN
1223  005130     000  $MFLG:  .BYTE   0               ;;MESSG. FLAG
1224  005131     000  $LFLG:  .BYTE   0               ;;LOG FLAG
1225  005132     000  $FFLG:  .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:
                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
                                ;
005134  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
005136  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
005144  105777  174074          1$:     TSTB    @$TKS           ;;WAIT FOR
005150  100375                          BPL     1$              ;;A CHARACTER
005152  117766  174070  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
005160  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
005166  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
005174  001013                          BNE     3$              ;;BRANCH IF NO
005176  105777  174042          2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
005202  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
005204  117746  174036                  MOVB    @$TKB,-(SP)     ;;GET CHARACTER
005210  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
005214  026627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
005220  001366                          BNE     2$              ;;IF NOT DISCARD IT
005222  000750                          BR      1$              ;;YES, RESUME
005224  026627  000004  000140  3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
005232  002407                          BLT     4$              ;;BRANCH IF YES
005234  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
005242  003003                          BGT     4$              ;;BRANCH IF YES
005244  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
005252  000002                  4$:     RTI                     ;;GO BACK TO USER
                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                ;*CALL:
                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S

005254  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
005256  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
005260  012703  005510          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
005264  022703  005517          2$:     CMP     #$TTYIN+7,R3    ;;BUFFER FULL?
005270  101456                          BLOS    4$              ;;BR IF YES
005272  104402                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
005274  112123                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
005276  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
005302  001022                          BNE     5$              ;;BR IF NO
005304  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
005306  001007                          BNE     6$              ;;BR IF NO
005310  112737  000134  005506          MOVB    #'\,9$          ;;TYPE A BACK SLASH
```

```
                                        TYPE    #9$
                                        MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
                                6$:     DEC     R3              ;;BACKUP BY ONE
                                        CMP     R3,#STTYIN      ;;STACK EMPTY?
                                        BLO     4$              ;;BR IF YES
                                        MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
                                        TYPE    #9$             ;;GO TYPE
                                        BR      2$              ;;GO READ ANOTHER CHAR.
                                5$:     TST     (SP)            ;;RUBOUT KEY SET?
                                        BEQ     7$              ;;BR IF NO
                                        MOVB    #'\,9$          ;;TYPE A BACK SLASH
                                        TYPE    #9$
                                        CLR     (SP)            ;;CLEAR THE RUBOUT KEY
                                7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
                                        BNE     8$              ;;BR IF NO
                                        TYPE    #CNTLU          ;;TYPE A CONTROL "U"
                                        BR      1$              ;;GO START OVER
                                8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
                                        BNE     3$              ;;BRANCH IF NO
                                        CLRB    (R3)            ;;CLEAR THE CHARACTER
                                        TYPE    #CRLF           ;;TYPE A "CR" & "LF"
                                        TYPE    #STTYIN         ;;TYPE THE INPUT STRING
                                        BR      2$              ;;GO PICKUP ANOTHER CHACTER
                                4$:     TYPE    #QUES           ;;TYPE A '?'
                                        BR      1$              ;;CLEAR THE BUFFER AND LOOP
                                3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
                                        TYPE    #9$
                                        CMPB    #15,(R3)+       ;;CHECK FOR RETURN
                                        BNE     2$              ;;LOOP IF NOT RETURN
                                        CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
                                        TYPE    #LF             ;;TYPE A LINE FEED
                                        TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
                                        MOV     (SP)+,R3        ;;RESTORE R3
                                        MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
                                        MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
                                        MOV     #STTYIN,4(SP)
                                        RTI                     ;;RETURN
                                9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
                                        .BYTE   0               ;;TERMINATOR
                                STTYIN: .BLKB   7               ;;RESERVE 7 BYTES FOR TTY INPUT
                                CNTLU:  .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
                                CNTLG:  .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
                                SMSWR:  .ASCIZ  <15><12>/SWR = /
                                SMNEW:  .ASCIZ  /  NEW = /

                                        .EVEN
                                .SBTTL  READ AN OCTAL NUMBER FROM THE TTY

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

```
                                        ;*CALL:
                                        ;*      RDOCT                   ;;READ AN OCTAL NUMBER
                                        ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
                                        ;*                              ;;HIGH ORDER BITS ARE IN SHIOCT
                                        ;
016746                  SRDOCT: MOV     (SP),-(SP)              ;;PROVIDE SPACE FOR THE
016666  000004  000002          MOV     4(SP),2(SP)             ;;INPUT NUMBER
010046                          MOV     R0,-(SP)                ;;PUSH R0 ON STACK
010146                          MOV     R1,-(SP)                ;;PUSH R1 ON STACK
010246                          MOV     R2,-(SP)                ;;PUSH R2 ON STACK
104463                  1$:     RDLIN                           ;;READ AN ASCIZ LINE
012600                          MOV     (SP)+,R0                ;;GET ADDRESS OF 1ST CHARACTER
010067  005702                  MOV     R0,5$                   ;;AND SAVE IT
005001                          CLR     R1                      ;;CLEAR DATA WORD
005002                          CLR     R2
112046                  2$:     MOVB    (R0)+,-(SP)             ;;PICKUP THIS CHARACTER
001420                          BEQ     3$                      ;;IF ZERO GET OUT
122716  000060                  CMPB    #'0,(SP)                ;;MAKE SURE THIS CHARACTER
003026                          BGT     4$                      ;;IS AN OCTAL DIGIT
122716  000067                  CMPB    #'7,(SP)
002423                          BLT     4$
006301                          ASL     R1                      ;;*2
006102                          ROL     R2
006301                          ASL     R1                      ;;*4
006102                          ROL     R2
006301                          ASL     R1                      ;;*8
006102                          ROL     R2
042716  177770                  BIC     #^C7,(SP)               ;;STRIP THE ASCII JUNK
062601                          ADD     (SP)+,R1                ;;ADD IN THIS DIGIT
000756                          BR      2$                      ;;LOOP
005726                  3$:     TST     (SP)+                   ;;CLEAN TERMINATOR FROM STACK
010166  000012                  MOV     R1,12(SP)               ;;SAVE THE RESULT
010267  005712                  MOV     R2,$HIOCT
012602                          MOV     (SP)+,R2                ;;POP STACK INTO R2
012601                          MOV     (SP)+,R1                ;;POP STACK INTO R1
012600                          MOV     (SP)+,R0                ;;POP STACK INTO R0
000002                          RTI                             ;;RETURN
005726                  4$:     TST     (SP)+                   ;;CLEAN PARTIAL FROM STACK
105010                          CLRB    (R0)                    ;;SET A TERMINATOR
104401                          TYPE                            ;;TYPE UP THRU THE BAD CHAR.
000000                  5$:     .WORD   0
104401  001312                  TYPE    $QUES                   ;;"?" "CR" & "LF"
000730                          BR      1$                      ;;TRY AGAIN
000000                  $HIOCT: .WORD   0                       ;;HIGH ORDER BITS GO HERE
                                ;
                                ;   INPUT OCTAL NUMBER ROUTINE
                                ;-------------------------------
                                ;
010546                  $INPUT: MOV     R5,-(SP)                ; SAVE REGISTER R5.
016605  000002                  MOV     2(SP),R5                ; GET FIRST PARAMETER ADDRESS.
012537  005760                  MOV     (R5)+,WHAT              ; GET MESSAGE ADDRESS.
012537  005740                  MOV     (R5)+,LOLIM             ; GET LOW LIMIT FOR THE  #
012537  005742                  MOV     (R5)+,HILIM             ; GET HIGH LIMIT FOR THE #.
012537  005744                  MOV     (R5)+,WHERE             ; GET ADDRESS OF INBUFFER
112537  005746                  MOVB    (R5)+,LOBITS            ; GET LOWMASK BITS.
112537  005747                  MOVB    (R5)+,ADRCNT            ; GET # OF #'S TO BE GENERATED.
```

```
1401  006252  010566  000002          MOV     R5,2(SP)        ;SAVE THE RETURN ADDRESS.
1402  006256  104401          INLP1:  TYPE                    ;TYPE THE MESSAGE.
1403  006260  000000          WHAT:   .WORD   0
1404  006262  104414                  RDOCT                   ;READ OCTAL # FROM KEYBOARD.
1405  006264  021627  006042          CMP     (SP),HILIM      ;IS IT IN HIGH LIMIT?
1406  006270  003003                  BGT     2$              ;BRANCH IF NO.
1407  006272  021627  006040          CMP     (SP),LOLIM      ;IS IT MORE THAN LOW LIMIT.
1408  006276  002005                  BGE     3$              ;BRANCH IF YES.
1409  006300  104401  001312  2$:     TYPE    ,QUES           ;TYPE "?"
1410  006304  104401  001313          TYPE    ,SCRLF          ;TYPE (CR),(LF)
1411  006310  000762                  BR      INLP1
1412  006312  013705  006[illegible]  3$:     MOV     WHERE,R5        ;GET BUFFER ADDRESS.
1413  006316  011625          4$:     MOV     (SP),(R5)+      ;SAVE THE # IN RIGHT PLACE.
1414  006320  062716  000[illegible]          ADD     #2,(SP)         ;NEXT SEQUENTIAL NUMBER.
1415  006324  105337  006047          DECB    ADACNT          ;COUNT BY 1.
1416  006330  001372                  BNE     4$              ;BRANCH IF NOT DONE.
1417  006332  005726                  TST     (SP)+           ;POP THE STACK POINTER.
1418  006334  012605                  MOV     (SP)+,R5        ;POP THE REG.5
1419  006336  000002                  RTI
1420  006340  000000  LOLIM:  .WORD   0
1421  006342  000000  HILIM:  .WORD   0
1422  006344  000000  WHERE:  .WORD   0
1423  006346  000     LOBITS: .BYTE   0
1424  006347  000     ADACNT: .BYTE   0
1425
1426                          ;  ADVANCE TO NEXT TEST HANDLER
1427                          ;-------------------------------
1428                          ;
1429  006350  013716  001442  .ADVANCE:       MOV     NEXT,(SP)               ; CRUNCH STACK WITH ADDRESSOF SCOPE CALL
1430  006354  005037  001444          CLR     LOCK                    ; RESET TIGHT LOOP ADDRESS
1431  006360  000002                  RTI                             ; CHECK TO SEE IF OLD TEST GETS REPEATED
1432
1433                          ;SAVE PC OF TEST THAT FAILED AND R0-R5
1434                          ;-------------------------------------
1435
1436  006362  016637  000004  001460  .SAVOS: MOV     4(SP),SAVPC     ;SAVE R7 (PC)
1437
1438                          ;SAVE R0-R5
1439
1440  006370  010537  001274          SVOS:   MOV     R5,$REG5        ;SAVE R5
1441  006374  010437  001272                  MOV     R4,$REG4        ;SAVE R4
1442  006400  010337  001270                  MOV     R3,$REG3        ;SAVE R3
1443  006404  010237  001266                  MOV     R2,$REG2        ;SAVE R2
1444  006410  010137  001264                  MOV     R1,$REG1        ;SAVE R1
1445  006414  010037  001262                  MOV     R0,$REG0        ;SAVE R0
1446  006420  000002                  RTI                             ;LEAVE.
1447
1448                          ;RESTORE R0-R5
1449
1450  006422  013700  001262          .RESOS: MOV     $REG0,R0        ;RESTORE R0
1451  006426  013701  001264                  MOV     $REG1,R1        ;RESTORE R1
1452  006432  013702  001266                  MOV     $REG2,R2        ;RESTORE R2
1453  006436  013703  001270                  MOV     $REG3,R3        ;RESTORE R3
1454  006442  013704  001272                  MOV     $REG4,R4        ;RESTORE R4
1455  006446  013705  001274                  MOV     $REG5,R5        ;RESTORE R5
1456  006452  000002                  RTI                             ;LEAVE
```

```
1457
1458                                    ;       ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
1459                                    ;       ;-----------------------------------------------------
1460                                    ;
1461  006154  104401  001313            .CONVR: TYPE    $CRLF
1462  006160  010046                    .CNVRT: MOV     R0,-(SP)
1463  006162  010146                            MOV     R1,-(SP)
1464  006164  010346                            MOV     R3,-(SP)
1465  006166  010446                            MOV     R4,-(SP)
1466  006170  010546                            MOV     R5,-(SP)
1467  006172  017601  000012                    MOV     @12(SP),R1
1468  006176  062766  000002  000012            ADD     #2,12(SP)
1469  006204  012137  006376                    MOV     (R1)+,WRDCNT
1470  006210  112137  006400            1$:     MOVB    (R1)+,CHRCNT
1471  006214  112137  006401                    MOVB    (R1)+,SPACNT
1472  006220  013137  006402                    MOV     @(R1)+,BINWRD
1473  006224  122737  000003  006400            CMPB    #3,CHRCNT
1474  006232  001003                            BNE     2$
1475  006234  042737  177400  006402            BIC     #177400,BINWRD
1476  006242  013704  006402            2$:     MOV     BINWRD,R4
1477  006246  113705  006400                    MOVB    CHRCNT,R5
1478  006252  012700  011112                    MOV     #TEMP,R0
1479  006256  010403                    3$:     MOV     R4,R3
1480  006260  042703  177770                    BIC     #177770,R3
1481  006264  062703  000060                    ADD     #060,R3
1482  006270  110320                            MOVB    R3,(R0)+
1483  006272  000241                            CLC
1484  006274  006004                            ROR     R4
1485  006276  000241                            CLC
1486  006300  006004                            ROR     R4
1487  006302  000241                            CLC
1488  006304  006004                            ROR     R4
1489  006306  005305                            DEC     R5
1490  006310  001365                            BNE     3$
1491  006312  012703  011154                    MOV     #MDATA,R3
1492  006316  114023            4$:     MOVB    -(R0),(R3)+
1493  006320  105337  006400                    DECB    CHRCNT
1494  006324  001374                            BNE     4$
1495  006326  105737  006401                    TSTB    SPACNT
1496  006332  001405                            BEQ     6$
1497  006334  112723  000040            5$:     MOVB    #040,(R3)+
1498  006340  105337  006401                    DECB    SPACNT
1499  006344  001373                            BNE     5$
1500  006346  105013                    6$:     CLRB    (R3)
1501  006350  104401  011154                    TYPE    ,MDATA
1502  006354  005337  006376                    DEC     WRDCNT
1503  006360  001313                            BNE     1$
1504  006362  012605                            MOV     (SP)+,R5
1505  006364  012604                            MOV     (SP)+,R4
1506  006366  012603                            MOV     (SP)+,R3
1507  006370  012601                            MOV     (SP)+,R1
1508  006372  012600                            MOV     (SP)+,R0
1509  006374  000002                            RTI
1510  006376  000000            WRDCNT: 0
1511  006400  000000            CHRCNT: 0
1512  006401                    SPACNT=CHRCNT+1
```

```
                        READ AN OCTAL NUMBER FROM THE TTY

1513  006402  000000    BINNRD: 0


                              ;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.

1529  006404  010046    $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
1530  006406  016600  000002          MOV     2(SP),R0        ;;GET TRAP ADDRESS
1531  006412  005740                  TST     -(R0)           ;;BACKUP BY 2
1532  006414  111000                  MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
1533  006416  006300                  ASL     R0              ;;POSITION FOR INDEXING
1534  006420  016000  006440          MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
1535  006424  000200                  RTS     R0              ;;GO TO ROUTINE
1536
1537
1538                        ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
1539
1540  006426  011646    $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
1541  006430  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
1542  006436  000002                  RTI                     ;;RESTORE THE PSW
1543
1544                        .SBTTL  TRAP TABLE
1545
1546                        ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
1547                        ;*BY THE "TRAP" INSTRUCTION.
1548
1549                        ;       ROUTINE
1550                        ;       -------
1551  006440  006426    $TRPAD: .WORD   $TRAP2
1552  006442  004404            $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
1553
1554
1555  006444  005134            $RDCHR  ;;CALL=RDCHR    TRAP+2(104402)  TTY TYPEIN CHARACTER ROUTINE
1556  006446  005254            $RDLIN  ;;CALL=RDLIN    TRAP+3(104403)  TTY TYPEIN STRING ROUTINE
1557  006450  005554            $RDOCT  ;;CALL=RDOCT    TRAP+4(104404)  READ AN OCTAL NUMBER FROM TTY
1558  006452  004360            .SCOPI  ;;CALL=SCOPI    TRAP+5(104405)  CALL TO LOOP ON CURRENT DATA HANDLER
1559  006454  006062            .SAVDS  ;;CALL=SAVDS    TRAP+6(104406)  CALL TO REGISTER SAVE ROUTINE
1560  006456  006122            .RESDS  ;;CALL=RESDS    TRAP+7(104407)  CALL TO REGISTER RESTORE ROUTINE
1561  006460  007334            .MSTCLR ;;CALL=MSTCLR   TRAP+10(104410) CALL TO ISSUE A MASTER CLEAR
1562  006462  007304            .DELAY  ;;CALL=DELAY    TRAP+11(104411) CALL TO DELAY
1563  006464  007352            .ROMCLK ;;CALL=ROMCLK   TRAP+12(104412) CALL TO CLOCK ROM ONCE
1564  006466  007420            .DATACLK        ;;CALL=DATACLK  TRAP+13(104413) CALL TO CLOCK DATA
1565  006470  007464            .TIMER  ;;CALL=TIMER    TRAP+14(104414) CALL TO DELAY A CLOCK TICK
1566  006472  005714            $INPUT  ;;CALL=INPUT    TRAP+15(104415) CALL TO OCTAL # INPUT ROUTINE
1567  006474  006154            .CONVRT ;;CALL=CONVRT   TRAP+16(104416) CALL TO    .....
1568  006476  006160            .CNVRT  ;;CALL=CNVRT    TRAP+17(104417) CALL TO    .......
```

```
                        .ADVANCE        ;;CALL=ADVANCE  TRAP+20(104420) CALL TO ADVANCE TO NEXT TEST
                ;
                ;-------------------------------------------------------------
                ;:*************************************************************
                        ;ERROR HANDLER
                        ;-------------
006500 006050
006502 004737 011216    SERROR: JSR     PC,CKSWR        ;CHECK FOR SOFT SWR
006506 032777 010000 172524     BIT     #SW12,@SWR      ;BELL ON ERROR?
006514 001406                   BEQ     XBX             ;BR IF NO BELL
006516 105777 172526            TSTB    @STPS           ;TTY READY.
006522 100003                   BPL     XBX             ;DON'T WAIT IF TTY NOT READY.
006524 112777 000207 172520     MOVB    #207,@STPB      ;PUSH A BELL AT THE TTY.
006532 032777 020000 172500     XBX:    BIT     #SW13,@SWR      ;DELETE ERROR PRINT OUT?
006540 001107                   BNE     HALTS           ;BR IF NO PRINT OUT WANTED.
006542 021637 001216            CMP     (SP),SERRPC     ;WAS THIS ERROR FOUND LAST TIME?
006546 001404                   BEQ     1$              ;BR IF YES
006550 011637 001216            MOV     (SP),SERRPC     ;RECORD BEING HERE
006554 105037 001203            CLRB    SERFLG          ;PREPARE HEADER
006560 104406                   SAVDS                   ;SAVE ALL PROC REGISTERS
006562 011605           1$:     MOV     (SP),R5         ;GET THE PC OF ERROR
006564 162705 000002            SUB     #2,R5           ;GET ADDRESS OF TRAP CALL
006570 011504                   MOV     (R5),R4         ;GET ERROR INSTRUCTION
006572 110437 001214            MOVB    R4,$ITEMB       ; COPY ERROR # FOR APT HANDLING
006576 006304                   ASL     R4              ;MULT BY TWO
006600 061504                   ADD     (R5),R4         ;DOUBLE IT
006602 006304                   ASL     R4              ;MULT AGAIN
006604 042704 177001            BIC     #177001,R4      ;CLEAR JUNK
006610 062704 001512            ADD     #SERRTB,R4      ;GET POINTER
006614 012437 006730            MOV     (R4)+,ERRMSG    ;GET ERROR MESSAGE
006620 012437 006742            MOV     (R4)+,DATAHD    ;GET DATA HEADRER
006624 011437 006754            MOV     (R4),DATABP     ;GET DATA TABLE
006630 105737 001203            TSTB    SERFLG          ;TYPE HEADREER
006634 001403                   BEQ     TYPMSG          ;BR IF YES
006636 005737 006754            TST     DATABP          ;DOES DATA TABLE EXIST?
006642 001040                   BNE     TYPDAT          ;BR IF YES.
006644 104401 001313    TYPMSG: TYPE    ,$CRLF
006650 104401 001313            TYPE    ,$CRLF
006654 005737 001444            TST     LOCK
006660 001402                   BEQ     1$
006662 104401 007767            TYPE    ,MASTEK
006666 104401 007755    1$:     TYPE    ,MTSTN
006672 104417 007072            CNVRT   ,XTSTN          ;SHOW IT
006676 104401 010044            TYPE    ,MERRPC         ;TYPE PC.
006702 104417 007064            CNVRT   ,ERTABO         ;SHOW IT
006706 104401 001313            TYPE    ,$CRLF          ;GIVE A CR/LF
006712 112737 177777 001203     MOVB    #-1,SERFLG      ;NO MORE HEADER UNLESS NO DATA TABLE.
006720 005737 006730            TST     ERRMSG          ;IS THERE AN ERROR MESSAGE?
006724 001402                   BEQ     WRKO.FM         ;BR IF NO.
006726 104401                   TYPE                    ;TYPE
006730 000000           ERRMSG: 0                       ;    ERROR MESSAGE
006732                  WRKO.FM:
006732 005737 006742            TST     DATAHD          ;DATA HEADER?
006736 001402                   BEQ     TYPDAT          ;BR IF NO
006740 104401                   TYPE                    ;TYPE
006742 000000           DATAHD: 0                       ;    DATA HEADER
```

```
                                ;DATA TABLE?
                                ;BR IF NO.
                                ;SHOW
                                ;    DATA TABLE
                                ;RESTORE PROC REGISTERS
                                ; IS APT RUNNING ?
                                ; SKIP APT CALL IF NOT.
                                ; COPY ERROR #.
                                ; CALL APT SERVICES.
                                ; ERROR # GOES HERE.
                                ; LOCK HERE.
                                ;IF ACT-11 AUTOMATIC MODE, HALT!!
                                ;HALT ON ERROR?
                                ;BR IF NO HALT ON ERROR
                                ;SAVE R0
                                ;SHOW ERROR PC IN DATA LIGHTS
                                ;HALT
                                ;GET R0
                                ;UPDATE ERROR COUNT
                                ;GOTO TOP OF TEST?
                                ;BR IF YES
                                ;GOTO NEXT TEST?
                                ;BR IF NO
                                ;SET FOR NEXT TEST
                                ;RESET SP
                                ;GOTO SPECIFIED TEST
                                ;SLPADR
                                ;ENTER HERE ON POWER FAILURE
                                ;-------------------------

                                .SBTTL  POWER DOWN AND UP ROUTINES

                                ;;**************************************************************
                                ;POWER DOWN ROUTINE
                                ;;SET FOR FAST UP
                                ;;PRIO:7
                                ;;PUSH R0 ON STACK
                                ;;PUSH R1 ON STACK
                                ;;PUSH R2 ON STACK
                                ;;PUSH R3 ON STACK
                                ;;PUSH R4 ON STACK
                                ;;PUSH R5 ON STACK
                                ;;PUSH @$WR ON STACK
                                ;;SAVE SP
                                ;;SET UP VECTOR
                                ;;HANG UP

                                ;**************************************************************
```

```
1681                                    ;POWER UP ROUTINE
1682  007152  012737  007270  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
1683  007160  013706  007274                  MOV     $SAVR6,SP        ;;GET SP
1684  007164  005037  007274                  CLR     $SAVR6           ;;WAIT LOOP FOR THE TTY
1685  007170  005237  007274          1$:     INC     $SAVR6           ;;WAIT FOR THE INC
1686  007174  001375                  BNE     1$               ;;OF  WORD
1687  007176  104401  007534                  TYPE    ,MPFAIL          ;
1688  007202  104417  007276                  CNVRT   ,PFTAB           ;
1689  007206  105037  001203                  CLRB    $ERFLG           ;CLEAR ERROR FLAG.
1690  007212  005037  001216                  CLR     $ERRPC           ; CLEAR LAST ERROR PC
1691  007216  013701  002066                  MOV     KMCSR,R1         ; RESTORE DEVICE ADDRESS.
1692  007222  005011                  CLR     (R1)             ; CLEAR THE CSR.
1693  007224  104410                  MSTCLR                   ;
1694  007226  012677  172006                  MOV     (SP)+,@SWR       ;POP STACK INTO @SWR
1695  007232  012605                  MOV     (SP)+,R5         ;POP STACK INTO R5
1696  007234  012604                  MOV     (SP)+,R4         ;POP STACK INTO R4
1697  007236  012603                  MOV     (SP)+,R3         ;POP STACK INTO R3
1698  007240  012602                  MOV     (SP)+,R2         ;POP STACK INTO R2
1699  007242  012601                  MOV     (SP)+,R1         ;POP STACK INTO R1
1700  007244  012600                  MOV     (SP)+,R0         ;POP STACK INTO R0
1701  007246  012737  007100  000024          MOV     #$PWRON,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
1702  007254  012737  000340  000026          MOV     #340,@#PWRVEC+2  ;;PRIO:7
1703  007262  104401                  TYPE                     ;;REPORT THE POWER FAILURE
1704  007264  007534          $PWRMG: .WORD   MPFAIL            ;;POWER FAIL MESSAGE POINTER
1705  007266  000002                  RTI
1706  007270  000000          $ILLUP: 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                  $TSTNM
1713
1714  007304                  .DELAY:
1715  007304  012777  000020  172562          MOV     #20,@KMP04
1716  007312  104412                  ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
1717  007314  121111                  121111                   ;POKE CLOCK DELAY BIT
1718  007316                  1$:
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     1$               ;BR IF NO
1723  007332  000002                  RTI
1724
1725  007334                  .MSTCLR:
1726  007334  152777  000100  172526          BISB    #BIT6,@KMCSRH    ;SET MASTER CLEAR
1727  007342  142777  000300  172520          BICB    #BIT6!BIT7,@KMCSRH  ;CLEAR MASTER CLEAR AND RUN
1728  007350  000002                  RTI                      ;RETURN
1729
1730  007352                  .ROMCLK:
1731  007352  152777  000002  172510          BISB    #BIT1,@KMCSRH    ;SET ROMI
1732  007360  013677  172512                  MOV     @(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     1$               ;BR IF SW06 =0
1736  007400  000000                  HALT                     ;HALT BEFORE CLOCKING INSTRUCTION
```

```
 1737  007402  152777  000003  172460  1$:     BISB    #BIT1!BIT0,@KMCSRH ;CLOCK INSTRUCTION
 1738  007410  142777  000007  172452          BICB    #BIT2!BIT1!BIT0,@KMCSRH  ;CLEAR ROMO, ROMI, STEP
 1739  007416  000002                          RTI
 1740
 1741  007420                                  .DATACLK:
 1742  007420  013637  011112                  MOV     @(SP)+,TEMP     ;PUT TICK COUNT IN TEMP
 1743  007424  062746  000002                  ADD     #2,-(SP)        ;ADJUST STACK
 1744  007430  152777  000020  172432  1$:     BISB    #BIT4,@KMCSRH   ;SET STEP LU
 1745  007436  027777  172424  172422          CMP     @KMCSR,@KMCSR   ;WASTE TIME
 1746  007444  142777  000020  172416          BICB    #BIT4,@KMCSRH   ;CLEAR STEP LU
 1747  007452  005337  011112                  DEC     TEMP            ;DEC TICK COUNT
 1748  007456  001364                          BNE     1$              ;BR IF NOT DONE
 1749  007460  000002                          RTI                     ;RETURN
 1750  007462  000001                  3$:     .BLKW 1
 1751
 1752  007464                                  .TIMER:
 1753  007464  013637  011112                  MOV     @(SP)+,TEMP     ;MOVE COUNT TO TEMP
 1754  007470  062746  000002                  ADD     #2,-(SP)        ;ADJUST STACK
 1755  007474                                  1$:
 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,@KMP04       ;IS PGM CLOCK BIT CLEAR?
 1759  007506  001772                          BEQ     1$              ;BR IF YES
 1760  007510                                  2$:
 1761  007510  104412                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 1762  007512  021364                          021364                  ;PORT4+IBUS# REG11
 1763  007514  032777  000002  172352          BIT     #2,@KMP04       ;IS PGM CLOCK BIT SET?
 1764  007522  001372                          BNE     2$              ;BR IF YES
 1765  007524  005337  011112                  DEC     TEMP            ;DEC COUNT
 1766  007530  001361                          BNE     1$              ;BR IF NOT DONE
 1767  007532  000002                          RTI                     ;RETURN
 1768
 1769  007534  050200  051127  043040  MPFAIL: .ASCIZ  <200>/PWR FAILED. RESTART AT TEST /
  (2)  007572  042600  042116  050040  MEPASS: .ASCIZ  <200>/END PASS DZKCA  /
  (2)  007614  051200  000     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  MCSRX:  .ASCIZ   /CSR: /
  (2)  007725  126     041505  020072  MVECX:  .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     000     MASTEK: .ASCIZ  /*/
  (2)  007771  200     042523  020124  MMEM:   .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>/           MAP OF KMC11 STATUS/
  (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)  010336  200     051503  0       CSR:    .ASCIZ  <200>/CSR ADDRESS?/
  (2)  010363  200     042526  0       VEC:    .ASCIZ  <200>/VECTOR ADDRESS?/
  (2)  010374  041200  020122  0       PRIO:   .ASCIZ  <200>/BR PRIORITY LEVEL? (4,5,6,7)?/
  (2)  010433  200     041122  0               .ASCIZ  <200>/WHICH LINE UNIT? IF NONE TYPE "N", IF M8201 TYPE "1", IF M8202 TYP
  (2)  010545  200     053523  0               .ASCIZ  <200>/SWITCH PAC#1 (DDCMP LINE #)?/
```

```
(2)  010603  000200  052523  052111  BM:     .ASCIZ  <200>/SWITCH PDP#2 (BM873 BOOT ADD)?/
(2)  010643  000200  051511  052040  COMM:   .ASCIZ  <200>/IS THE LOOP BACK CONNECTOR ON?/
(2)  010703  000200  047516  042040  NOACT:  .ASCIZ  <200>/NO DEVICES ARE SELECTED/
(2)  010734  100200  046513  020503  CONERR: .ASCIZ  <200><200>/KMC11 AT NONSTANDARD ADDRESS  PC: /
(2)  011001  000200  051105  042520  CHERR:  .ASCIZ  <200>/EXPECTED  FOUND/
(2)  011022  024040  046513  024503  KMCH:   .ASCIZ  / (KMC) /
(2)  011032  044600  046114  043505  MILINT: .ASCIZ  <200>/ILLEGAL INTERRUPT ERROR/
(2)          011064                  .EVEN
(2)  011064  000005                  XSTATQ: 5
1770 011066  006     003                    .BYTE   6,3
1771 011070  001276                  $TMP0
1772 011072  006     003                    .BYTE   6,3
1773 011074  001300                  $TMP1
1774 011076  006     003                    .BYTE   6,3
1775 011100  001302                  $TMP2
1776 011102  006     003                    .BYTE   6,3
1777 011104  001304                  $TMP3
1778 011106  006     002                    .BYTE   6,2
1779 011110  001306                  $TMP4
1780                                 .EVEN
1781
1782                                         ;BUFFERS FOR INPUT-OUTPUT
1783
1784 011112  000000                  TEMP:   0
1785         011154                  .=.+40
1786 011154  000000                  MDATA:  0
1787         011216                  .=.+40
1788
1789
1790                                         ;ROUTINE USED TO CHANGE SOFTWARE SWITCH
1791                                         ;REGISTER USING THE CONSOLE TERMINAL
1792                                         ;-----------------------------------
1793
1794 011216  022737  000176  001240  CKSWR:  CMP     @#SWREG,SWR     ;IS THE SOFT SWR BEING USED?
1795 011224  001075                          BNE     CKSWR5          ;BR IF NO
1796 011226  132737  000001  001336          BITB    #1,@#ENV        ; IS IT RUNNING UNDER APT?
1797 011234  001071                          BNE     CKSWR5          ; EXIT IF YES.
1798 011236  022777  000007  170002          CMP     #7,@#TKB        ;WAS CTRL G TYPED? (7 BIT ASCII)
1799 011244  001404                          BEQ     1$              ;BR IF YES
1800 011246  022777  000207  167772          CMP     #207,@#TKB      ;WAS CTRL G TYPED? (8 BIT ASCII)
1801 011254  001061                          BNE     CKSWR5          ;BR IF NO
1802 011256  010246                  1$:     MOV     R2,-(SP)        ;STORE R2
1803 011260  010346                          MOV     R3,-(SP)        ;STORE R3
1804 011262  010446                          MOV     R4,-(SP)        ;STORE R4
1805 011264  012737  177777  011422          MOV     #-1,SWFLG       ;SET SOFT TYPE OUT FLAG
1806 011272  005002                  CKSWR1: CLR     R2              ;CLEAR NEW SWR CONTENTS
1807 011274  012704  177777                  MOV     #-1,R4          ;SET FLAG TO ALL ONES
1808 011300  104401  005531                  TYPE    ,SWSWR          ;TYPE "SWR= "
1809 011304  104417                  CKSWR2: CNVRT                   ;TYPE OUT PRESENT CONTENTS
1810 011306  011456                          SOFTSW                  ;OF SOFT SWITCH REGISTER
1811 011310  104401  005542          CKSWR3: TYPE    ,SWNEW          ;TYPE "NEW? "
1812 011314  004737  011424          CKSWR4: JSR     PC,INCHAR       ;GET RESPONSE
1813 011320  022703  000015                  CMP     #15,R3          ;WAS IT A CR?
1814 011324  001424                          BEQ     5$              ;BR IF YES
1815 011326  022703  000012                  CMP     #12,R3          ;WAS IT A LF?
1816 011332  001416                          BEQ     4$              ;BR IF YES
```

```
1817  011334  022703  000025          CMP     #25,R3          ;WAS IT CTRL U?
1818  011340  001754                  BEQ     CKSWR1          ;BR IF YES(START OVER)
1819  011342  022703  000007          CMP     #7,R3           ;IF CNTL G GET NEXT CHAR
1820  011346  001753                  BEQ     CKSWR4
1821  011350  005004                  CLR     R4              ;IT MUST BE A DIGIT SO CLR FLAG
1822  011352  042703  177770          BIC     #177770,R3      ;ONLY 0-7 ARE LEGAL SO MASK OFF BITS
1823  011356  006302                  ASL     R2              ;SHIFT R2 3 TIMES
1824  011360  006302                  ASL     R2
1825  011362  006302                  ASL     R2
1826  011364  050302                  BIS     R3,R2           ;ADD LAST DIGIT
1827  011366  000752                  BR      CKSWR4          ;GET NEXT CHARACTER
1828  011370  012766  002402  000006  4$:     MOV     #.START,6(SP)   ;LF WAS TYPED SO GO TO START
1829  011376  005704          5$:     TST     R4              ;IS FLAG CLEAR?
1830  011400  001002                  BNE     6$              ;IF NOT DON'T CHANGE SOFT SWR
1831  011402  010277  167632          MOV     R2,@SWR         ;IF YES THEN WRITE NEW CONTENTS TO SOFT SWR
1832  011406  005037  011422          6$:     CLR     SWFLG           ;CLEAR TYPEOUT FLAG
1833  011412  012604                  MOV     (SP)+,R4        ;RESTORE R4
1834  011414  012603                  MOV     (SP)+,R3        ;RESTORE R3
1835  011416  012602                  MOV     (SP)+,R2        ;RESTORE R2
1836  011420  000207          CKSWR5: RTS     PC              ;RETURN
1837
1838  011422  000000          SWFLG:  0
1839
1840  011424  105777  167614  INCHAR: TSTB    @STKS
1841  011430  100375                  BPL     .-4
1842  011432  017703  167610          MOV     @STKB,R3
1843  011436  105777  167606          TSTB    @STPS
1844  011442  100375                  BPL     .-4
1845  011444  010377  167602          MOV     R3,@STPB
1846  011450  042703  000200          BIC     #BIT7,R3
1847  011454  000207                  RTS     PC
1848
1849  011456  000001          SOFTSW: 1
1850  011460  006     002             .BYTE   6,2
1851  011462  000176                  SWREG
```

```
                                        ;
                                        ;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.
                                        ;
011464 005737 001470        CYCLE:  TST     KMACTV          ;ARE ANY KMC11'S TO BE TESTED?
011470 001004                       BNE     1$              ;BR IF OK.
011472 104401 010703                TYPE    ,NOACT          ;NO KMC11'S SELECTED!!
011476 000000                       HALT                    ;STOP THE SHOW.
011500 000776                       BR      .-2             ;DISQUALIFY CONT. SW.
011502 000241               1$:     CLC                     ;CLEAR PROC. CARRY BIT.
011504 006137 001500                ROL     RUN             ;UPDATE POINTER
011510 006137 001500                ADC     RUN             ;CATCH CARRY FROM RUN
011514 062737 000004 001504         ADD     #4,MILK         ;UPDATE POINTER
011522 062737 000010 001502         ADD     #10,CREAM       ;UPDATE ADDRESS POINTER.
011530 027737 002300 001502         CMP     #KM.MAP+200,CREAM
011536 001006                       BNE     2$              ;KEEP GOING; NOT ALL TESTED FOR.
011540 012737 002100 001502         MOV     #KM.MAP,CREAM   ;RESET ADDRESS POINTER.
011546 012737 002302 001504         MOV     #CNT.MAP,MILK   ;RESET PASS COUNT POINTER
011554 033737 001500 001470 2$:     BIT     RUN,KMACTV      ;IS THIS ONE ACTIVE?
011562 001747                       BEQ     1$              ;BR IF NO
011564 013700 001502                MOV     CREAM,R0        ;GET ADDRESS POINTER
011570 013702 001504                MOV     MILK,R2         ;GET PASS COUNT POINTER
011574 012037 002066                MOV     (R0)+,KMCSR     ;LOAD SYSTEM CTRL. REG
011600 011037 002056                MOV     (R0),KMRVEC     ;LOAD VECTOR
011604 042737 177000 002056         BIC     #177000,KMRVEC  ;CLEAR UNWANTED BITS
011612 012037 002050                MOV     (R0)+,STAT1     ;LOAD STAT1
011616 012037 002052                MOV     (R0)+,STAT2     ;LOAD STAT2
011622 012037 002054                MOV     (R0)+,STAT3     ;LOAD STAT3
011626 012237 001324                MOV     (R2)+,PASS      ;LOAD PASS COUNT
011632 012237 001212                MOV     (R2)+,ERRTTL    ;LOAD ERROR COUNT
011636 012700 000002                MOV     #2,R0           ;SAVE CORE THIS WAY!
011642 013737 002066 002070         MOV     KMCSR,KMCSRH
011650 005237 002070                INC     KMCSRH
011654 013737 002070 002072         MOV     KMCSRH,KMCTL
011662 005237 002072                INC     KMCTL
011666 013737 002072 002074         MOV     KMCTL,KMP04
011674 060037 002074                ADD     R0,KMP04
011700 013737 002074 002076         MOV     KMP04,KMP06
011706 060037 002076                ADD     R0,KMP06

011712 013737 002056 002060         MOV     KMRVEC,KMRLVL   ;PTY LVL
011720 060037 002060                ADD     R0,KMRLVL       ;
011724 013737 002060 002062         MOV     KMRLVL,KMTVEC   ;TX VEC
011732 060037 002062                ADD     R0,KMTVEC       ;
011736 013737 002062 002064         MOV     KMTVEC,KMTLVL   ;TX LVL
011744 060037 002064                ADD     R0,KMTLVL       ;
011750 012777 034664 170100         MOV     #[illegible],@#[illegible]  ;SET UP INTERRUPT VECTORS.
011756 012777 000240 170074         MOV     #PR[illegible],@#[illegible]  ;SET PRIORITY 5.
011764 012777 035360 170070         MOV     #OU[illegible],@#KMTVEC  ;SET UP INTERRUPT VECTORS.
011772 012777 000240 170064         MOV     [illegible]             ;SET PRIORITY 5.
012000 012737 000200 177776         MOV     [illegible]             ;SET MAIN PROGRAM PRIORITY AS 4.
```

```
1908                                                                    ;SOTHAT KMC CAN INTERRUPT...
1909
1910  012006  032737  000002  001446          BIT     #SW01,STRTSW    ;IS TEST NO. SELECTED
1911  012014  001447                          BEQ     7$              ;BR IF NO
1912  012016                          4$:
1913  012016  005737  000042                  TST     @#42            ;RUNNING IN AUTO MODE?
1914  012022  001044                          BNE     7$              ;BR IF YES
1915  012024  104401  001313                  TYPE    ,$CRLF
1916  012030  104415                          INPUT
1917  012032  007755                          MTSTN
1918  012034  000001                          1
1919  012036  001000                          1000
1920  012040  001202                          $TSTNM
1921  012042      000                 .BYTE   0
1922  012043      001                 .BYTE   1
1923  012044  012700  013774                  MOV     #TST1,R0
1924  012050  022710                  5$:     CMP     (PC)+,(R0)      ;CMP FIRST WORD TO 12737
1925  012052  012737                          MOV     (PC)+,@(PC)+
1926  012054  001020                          BNE     6$              ;BR IF NOT SAME
1927  012056  023760  001202  000002          CMP     $TSTNM,2(R0)    ;DOES $TSTNM MATCH?
1928  012064  001014                          BNE     6$              ;BR IF NO
1929  012066  022760  001202  000004          CMP     #$TSTNM,4(R0)   ;IS LAST WORD OK?
1930  012074  001010                          BNE     6$              ;BR IF NO
1931  012076  010037  001206                  MOV     R0,$LPADR       ;IT IS A LEGAL TEST SO DO IT
1932  012102  104401  007614                  TYPE    ,MR
1933  012106  042737  000002  001446          BIC     #SW01,STRTSW
1934  012114  000412                          BR      8$
1935  012116  005720                  6$:     TST     (R0)+           ;POP R0
1936  012120  020027  034324                  CMP     R0,#TLAST+10    ;AT END YET?
1937  012124  001351                          BNE     5$              ;BR IF NO
1938  012126  104401  001312                  TYPE    ,$QUES          ;YES ILLEGAL TEST NO.
1939  012132  000731                          BR      4$              ;TRY AGAIN
1940
1941  012134  012737  013774  001206  7$:     MOV     #TST1,$LPADR    ;PREPARE $LPADR ADDRESS
1942  012142  013701  002066          8$:     MOV     KMCSR,R1        ;R1 = BASE KMC11 ADDRESS
1943  012146  000177  167034                  JMP     @$LPADR         ;GO START TESTING.
1944
1945
1946                                          ;ROUTINE USED TO "AUTO SIZE" THE KMC11
1947                                          ;CSR AND VECTOR.
1948                                          ;NOTE:  THE CSR MAY BE ANY WHERE IN THE FLOATING
1949                                          ;       ADDRESS RANGE (160000:164000)
1950                                          ;       AND THE VECTOR MAY BE ANY WHERE IN THE
1951                                          ;       FLOATING VECTOR RANGE (300:770)
1952                                          ;
1953
1954  012152                          AUTO.SIZE:
1955  012152  000005                          RESET                   ;INSURE A BUS INIT.
1956  012154  012702  002100          CSRMAP: MOV     #KM.MAP,R2      ;LOAD MAP POINTER.
1957  012160  005022                  1$:     CLR     (R2)+           ;ZERO ENTIRE MAP
1958  012162  022702  002300                  CMP     #KM.END,R2      ;ALL DONE?
1959  012166  001374                          BNE     1$              ;BR IF NO
1960  012170  005037  001472                  CLR     KMNUM           ;SET OCTAL NUMBER OF KMC11'S TO 0
1961  012174  012702  002100                  MOV     #KM.MAP,R2      ;R2 POINTS TO KMC MAP
1962  012200  005037  001470                  CLR     KMACTV          ;CLEAR ACTIVE
1963  012204  032737  000001  001446          BIT     #SW00,STRTSW    ;QUESTIONS?
```

```
                                        BNE     .+6             ;BR IF YES
012574                                  JMP     7$              ;IF NO SKIP QUESTIONS
000001  001306                          MOV     #1,STMP4        ;START WITH 1
                                        INPUT
                                        NUM
                                        1
                                        16.
                                        STMP2
                                        .BYTE   0
                                        .BYTE   1
001302  001472                          MOV     STMP2,KMNUM     ;KMNUM = HOW MANY
001313                  12$:            TYPE    ,SCRLF
                                        CONVRT                  ;TYPE WHICH KMC IS BEING DONE
                                        WHICH                   ;STMP4 IS WHICH KMC
001306                                  INC     STMP4
                                        INPUT
                                        CSR
                                        160000
                                        164000
                                        STMP3
                                        .BYTE   0
                                        .BYTE   1
001304                                  MOV     STMP3,(R2)+     ;STORE CSR IN MAP
                                        INPUT
                                        VEC
                                        0
                                        776
                                        STMP3
                                        .BYTE   0
                                        .BYTE   1
001304                                  MOV     STMP3,(R2)      ;STORE VECTOR IN MAP
                        10$:            TYPE
                                        PRIO                    ;ASK WHAT BR LEVEL
013520                                  JSR     PC,INTTY        ;GET RESPONSE
000024                                  CMP     #4,R3
                                        BHI     50$             ;BR IF LESS THAN 4
000027                                  CMP     #7,R3
                                        BLO     50$             ;BR IF GREATER THAN 7
000011                                  MOV     #11,R4          ;R4 = NUMBER OF SHIFTS
                                        ASL     R3              ;SHIFT R3 LEFT
                                        DEC     R4              ;DEC SHIFT COUNT
                                        BNE     .-4             ;BR IF NOT DONE
170777                                  BIC     #170777,R3      ;BIC UNWANTED BITS
                                        BIS     R3,(R2)         ;PUT BR LEVEL IN STATUS MAP
                                        BR      8$              ;CONTINUE
                        50$:            TYPE
                                        SQUES                   ;RESPONSE IS OUT OF LIMITS
                                        BR      10$             ;TRY AGAIN
                        8$:
                        9$:
                        16$:            TYPE
                                        MODU                    ;ASK WHICH LINE UNIT
013520                                  JSR     PC,INTTY        ;GET REPLY
000021                                  CMP     #21,R3          ;"1"
                                        BEQ     30$
000022                                  CMP     #22,R3          ;"2"
```

```
2020  012422  001412                  BEQ     31$
2021  012424  022703  000116          CMP     #116,R3         ;"N"
2022  012430  001403                  BEQ     32$
2023  012432  104401                  TYPE
2024  012434  001312                  SQUES                   ;IF NOT A 1,2 OR N TYPE "?"
2025  012436  000760                  BR      16$             ;TRY AGIAN
2026  012440  052722  010000  32$:    BIS     #BIT12,(R2)+    ;SET BIT 12 IN STAT2 IF NO LU
2027  012444  062222                  CMP     (R2)+,(R2)+     ;POP OVER STAT2 AND STAT3
2028  012446  000445                  BR      33$
2029  012450  052712  020000  31$:    BIS     #BIT13,(R2)     ;SET BIT 13 IN STAT2 IF MB202
2030  012454  104401  30$:    TYPE
2031  012456  010643                  CONN                    ;ASK IF LOOP-BACK IS ON
2032  012460  004737  013520          JSR     PC,INTTY        ;GET REPLY
2033  012464  022703  000131          CMP     #131,R3         ;Y
2034  012470  001406                  BEQ     17$
2035  012472  022703  000116          CMP     #116,R3         ;N
2036  012476  001406                  BEQ     18$
2037  012500  104401                  TYPE
2038  012502  001312                  SQUES                   ;IF NOT Y OR N TYPE "?"
2039  012504  000763                  BR      30$             ;TRY AGAIN
2040  012506  052722  040000  17$:    BIS     #BIT14,(R2)+    ;TURNAROUND IS CONNECTED
2041  012512  000402                  BR      19$
2042  012514  042722  040000  18$:    BIC     #BIT14,(R2)+    ;NO TURNAROUND
2043  012520                  19$:
2044  012520  104415                  INPUT
2045  012522  010545                  LINE
2046  012524  000000                  0
2047  012526  000377                  377
2048  012530  001304                  $TMP3
2049  012532      000                 .BYTE   0
2050  012533      001                 .BYTE   1
2051  012534  113722  001304          MOVB    $TMP3,(R2)+     ;STORE SWITCH PAC IN MAP
2052  012540  104415                  INPUT
2053  012542  010603                  BM
2054  012544  000000                  0
2055  012546  000377                  377
2056  012550  001304                  $TMP3
2057  012552      000                 .BYTE   0
2058  012553      001                 .BYTE   1
2059  012554  113722  001304          MOVB    $TMP3,(R2)+     ;STORE SWITCH PAC IN MAP
2060  012560  005722                  TST     (R2)+           ;POP OVER STAT3
2061  012562  005337  001302  33$:    DEC     $TMP2           ;DEC KMC COUNT
2062  012566  001230                  BNE     12$             ;BR IF MORE TO DO
2063  012570  000137  013126          JMP     13$             ;CONTINUE
2064  012574  012701  160000  7$:     MOV     #160000,R1      ;SET FOR FIRST ADDRESS TO BE TESTED
2065  012600  012737  013220  000004  MOV     #6$,@#4         ;SET FOR NON-EXISTANT DEVICE TIME OUT
2066  012606  005011          2$:     CLR     (R1)            ;CLEAR SEL0
2067  012610  005711                  TST     (R1)            ;IF KMC11 KMCSR S/B 0
2068  012612  001135                  BNE     3$              ;IF NO DEV ; TRAP TO 4. IF NO BIT 8 THEN NO KMC11
2069  012614  005061  000006          CLR     6(R1)           ;CLEAR SEL6
2070  012620  005761  000006          TST     6(R1)           ;IF KMC11 THEN KMRIC S/B =0!
2071  012624  001130                  BNE     3$              ;BR IF NOT KMC11
2072  012626  012711  002000          MOV     #BIT10,(R1)     ;SET ROMO
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!
```

```
                        CMP     #125252,4(R1)   ;WAS IT WRITTEN?
                        BNE     3$              ;IF NO IT IS NOT CRAM
                ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A KMC11 CSR ADDRESS.
                2$:
                22$:    MOV     R1,(R2)+        ;STORE CSR IN CORE TABLE.
                15$:    MOV     #BIT9,(R1)      ;CLEAR LINE UNIT LOOP
                        CLR     4(R1)           ;CLEAR PORT4
                        MOV     #122113,6(R1)   ;LOAD INSTRUCTION (CLR DTR)
                        BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION
                        MOV     #021264,6(R1)   ;LOAD INSTRUCTION
                        BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION
                        CMPB    #377,4(R1)      ;IS IT ALL ONES?
                        BNE     .+10            ;BR IF NO
                        BIS     #BIT12,(R2)     ;IF YES, NO LINE UNIT, SET STATUS BIT
                        BR      20$
                        BIT     #BIT1,4(R1)     ;IS SWITCH A ONE?
                        BEQ     .+10            ;BR IF MODE01
                        BIS     #BIT13!BIT14,(R2) ;MODE02 ASSUME CONNECTOR
                        BR      20$             ;CONNECTOR ON)
                        BIT     #BIT3,4(R1)     ;IS MODY SET
                        BNE     20$             ;BR IF MODE01 NO CONNECTOR (ON LINE)
                        MOV     #BIT6,4(R1)     ;LOAD PORT4
                        MOV     #122113,6(R1)   ;LOAD INSTRUCTION
                        BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION(SET DTR)
                        MOV     #021264,6(R1)   ;LOAD INSTRUCTION
                        BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION(READ MODEM REG)
                        BIT     #BIT3,4(R1)     ;IS MODY SET NOW?
                        BEQ     20$             ;BR IF NO CONNECTOR
                        BIS     #BIT14,(R2)     ;SET STATUS BIT FOR CONNECTOR
                20$:    TST     (R2)+           ;POP POINTER
                        MOV     #021324,6(R1)   ;PUT INSTRUCTION IN PORT6
                        MOV     #BIT9!BIT8,(R1) ;PORT4=LU 15
                        BISB    4(R1),(R2)+     ;STORE DDCMP LINE # IN TABLE
                        MOV     #021344,6(R1)   ;PORT6=INSTRUCTION
                        MOV     #BIT8!BIT9,(R1) ;CLOCK INSTR.
                        BISB    4(R1),(R2)+     ;STORE MODEM ADD IN TABLE
                        TST     (R2)+           ;POP OVER STAT3
                        CLR     (R1)            ;CLEAR MOM1
                        INC     KMNUM           ;UPDATE DEVICE COUNTER
                        CMP     #20,KMNUM       ;ARE MAX. NO. OF DEV FOUND?
                        BEQ     13$             ;YES DON'T LOOK FOR ANY MORE.
                3$:     CLR     (R1)            ;CLEAR BIT 10
                        CLR     6(R1)           ;CLEAR SEL 6
                14$:    ADD     #10,R1          ;UPDATE CSR POINTER ADDRESS
                        CMP     #164000,R1
                        BNE     2$              ;BR IF MORE ADDRESS TO CHECK.
                13$:    CLR     KMACTV
                        TST     KMNUM           ;WERE ANY KMC11'S FOUND AT ALL?
                        BEQ     5$              ;ERROR AUTO SIZER FOUND NO KMC11'S IN THIS SYS.
                        MOV     KMNUM,R1
                        MOV     R1,SAVNUM       ;SAVE NUMBER OF DEVICES
                4$:     CLC
                        ROL     KMACTV          ;GENERATE ACTIVE REGISTER OF DEVICES.
                        INC     KMACTV          ;SET THE BIT
                        DEC     R1
                        BNE     4$              ;BR IF MORE TO GENERATE
```

```
2132  013166  012737  000006  000024          MOV     #6,@#24         ;RESTORE TRAP VECTOR
2133  013174  013737  001470  001474          MOV     KMACTV,SAVACT   ;SAVE ACTIVE REGISTER
2134  013202  000137  013234                  JMP     VECMAP          ;GO FIND THE VECTOR NOW.
2135  013206  104401  007617          5$:     TYPE    ,MERR2          ;NOTIFY OPR THAT NO KMC11'S FOUND.
2136  013212  005000                          CLR     R0              ;MAKE DATA LIGHTS ZERO
2137  013214  000000                          HALT                    ;STOP THE SHOW
2138  013216  000776                          BR      .-2             ;DISABLE CONT. SW.
2139  013220  012716  013114          6$:     MOV     #14$,(SP)       ;ENTERED BY NON-EXISTANT TIME-OUT.
2140  013224  000002                          RTI                     ;RETURN TO MAINSTREAM
2141
2142  013226  000001                  SWITCH: 1
2143  013230     002          0               .BYTE   2,2
2144  013232  001306                          STMP4
2145
2146  013234  032737  000001  001     VECMAP: BIT     #SW00,STRTSW
2147  013242  001114                          BNE     5$
2150  013244  012737  000340  000     1$:     MOV     #340,@#22       ;SET IOT TRAP PRIO TO 7
2151  013252  012737  013426  000020          MOV     #14$,@#20       ;SET IOT TRAP VECTOR
2152  013260  012702  002100                  MOV     #KM.MAP,R2      ;SET SOFTWARE POINTER
2153  013264  012700  000300                  MOV     #300,R0         ;FLOATING VECTORS START HERE.
2154  013270  012701  000302                  MOV     #302,R1         ;PC OF IOT INSTR.
2155  013274  010120                  1$:     MOV     R1,(R0)+        ;START FILLING VECTOR AREA
2156  013276  012721  000004                  MOV     #4,(R1)+        ;WITH .+2; IOT
2157  013302  060021                          CMP     (R0)+,(R1)+     ;ADD 2 TO R0 +R1
2160  013304  020127  001000                  CMP     R1,#1000
2161  013310  101771                          BLOS    1$              ;BR IF MORE TO FILL
2162  013312  013737  001470  001276          MOV     KMACTV,STMP0    ;STORE TEMPORALLY
2163  013320  006037  001276          2$:     ROR     STMP0           ;BRING OUT A BIT
2164  013324  103003                          BCC     3$              ;BR IF ALL DONE
2165  013326  012704  000012                  MOV     #12,R4          ;R4 IS INDEX REGISTER
2166  013332  016437  013504  177776          MOV     BRLVL(R4),PS    ;SET PS TO 7
2167  013340  011201                          MOV     (R2),R1
2170  013342  012761  000200  000004          MOV     #200,4(R1)
2171  013350  012711  001000                  MOV     #BIT9,(R1)      ;SET ROMI
2172  013354  012761  121111  000006          MOV     #121111,6(R1)   ;PUT INSTRUCTION IN PORT6
2173  013362  012711  001400                  MOV     #BIT9!BIT8,(R1) ;FORCE AN INTERRUPT
2174  013366  105200                  7$:     INCB    R0              ;STALL
2175  013370  001376                          BNE     .-2             ;FOR TIME TO INTERUPT
2176  013372  162704  000002                  SUB     #2,R4           ;GET NEXT LOWEST PS LEVEL
2177  013376  001404                          BEQ     6$              ;BR IF R4 = 0
2200  013400  016437  013504  177776          MOV     BRLVL(R4),PS    ;MOVE NEXT LOWER LEVEL IN PS
2201  013406  000767                          BR      7$              ;BR TO DELAY
2202  013410  052762  005300  000002  6$:     BIS     #5300,2(R2)     ;NO INTERUPT ASSUME 300 AT LEVEL 5 AND FIX KMC11 LATER
2203  013416  005011          3$:     CLR     (R1)                    ;CLEAR ROMI
2204  013420  062702  000010                  ADD     #10,R2          ;POP SOFTWARE POINTER
2205  013424  000736                          BR      2$              ;KEEP GOING
2206  013426  052662  000002          4$:     BIS     (SP),2(R2)      ;GET VECTOR ADDRESS
2207  013432  042762  000007  000002          BIC     #7,2(R2)        ;CLEAR JUNK
2210  013440  016405  013506                  MOV     BRLVL+2(R4),R5  ;GET BR LEVEL OF KMC11
2211  013444  006305                          ASL     R5              ;SHIFT LEVEL 4 PLACES
2212  013446  006305                          ASL     R5              ;TO THE LEFT FOR THE
2213  013450  006305                          ASL     R5              ;STATUS TABLE
2214  013452  006305                          ASL     R5
2215  013454  042705  170777                  BIC     #170777,R5      ;CLEAR UNWANTED BITS
2216  013460  050562  000002                  BIS     R5,2(R2)        ;PUT BR LEVEL IN STATUS TABLE
2217  013464  022626                          CMP     (SP)+,(SP)+     ;POP IOT JUNK OFF STACK
```

```
013466 012716 013416                    MOV     #3$,(SP)        ;SET FOR RETURN
013472 000002                           RTI
013474 012737 004134 000020     5$:     MOV     #$SCOPE,@#20    ; RESTORE SCOPE VECTOR
013502 000207                           RTS     PC              ;ALL DONE WITH "AUTO SIZING"

013504 000000                   BRLVL:  PR0     ;LEVEL 0
013506 000000                           PR0     ;LEVEL 0
013510 000200                           PR4     ;LEVEL 4
013512 000240                           PR5     ;LEVEL 5
013514 000300                           PR6     ;LEVEL 6
013516 000340                           PR7     ;LEVEL 7


013520 105777 165520            INTTY:  TSTB    @#TKS           ;WAIT FOR DONE
013524 100375                           BPL     .-4
013526 017703 165514                    MOV     @#TKB,R3        ;PUT CHAR IN R3
013532 105777 165512                    TSTB    @#TPS           ;WAIT UNTIL PRINTER IS READY
013536 100375                           BPL     .-4
013540 010377 165506                    MOV     R3,@#TPB        ;ECHO CHAR
013544 042703 000240                    BIC     #BIT7!BIT5,R3   ;MASK OFF LOWER CASE
013550 000207                           RTS     PC              ;RETURN

013552                          APT.SIZE:
013552 000005                           RESET
013554 010046                           MOV     R0,-(SP)        ;;PUSH R0 ON STACK
013556 010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
013560 010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
013562 010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
013564 005037 013766                    CLR     VECTR           ;CLEAR THE LOCAL VARIABLE
013570 005037 013772                    CLR     PRIRTY          ;CLEAN UP LOCAL VARIABLE
013574 013700 001376                    MOV     SCOM1,R0        ;GET THE DEVICE COUNT
013600 010037 001476                    MOV     R0,SAVNUM       ;SAVE THE NO. OF DEVICES
013604 012701 001346                    MOV     #KWMS1,R1       ;GET EXTRA INFO, BITS POINTER
013610 013737 001372 013770             MOV     $BASE,BASE      ;GET BASE CSR ADDRESS
013616 113737 001366 013766             MOVB    $VECT1,VECTR    ;GET THE VECTOR
013624 113737 001367 013772             MOVB    $VECT1+1,PRIRTY ;GET THE PRIORITY
013632 013737 001374 001470             MOV     $DEVM,KWACTV    ;SAVE THE KMC'S SELECTED ACTIVE
013640 013737 001470 001474             MOV     KWACTV,SAVACT   ;SAVE THE ACTIVE REGISTER
013646 012702 001402                    MOV     #SDDWD,R2       ;GET ADDRESS OF FIRST DEVICE DESCRIPTOR WORD
013652 012703 002100                    MOV     #KM.MAP,R3      ;GET POINTER TO DEVICE MAP
013656 005023                   3$:     CLR     (R3)+           ;CLEAR DEVICE MAP
013660 022703 002300                    CMP     #KM.END,R3      ;IS WHOLE DEV.MAP CLEARED?
013664 003374                           BGT     3$              ;NO, THEN GO ON.
013666 012703 002100                    MOV     #KM.MAP,R3      ;RESTORE DEV.MAP POINTER.
013672 013723 013770            1$:     MOV     BASE,(R3)+      ;LOAD CSR ADDRESS
013676 112163 000001                    MOVB    (R1)+,1(R3)     ;GET EXTRA INFO. BITS
013702 006213                           ASR     (R3)            ;SET IT IN RIGHT POSITION.
013704 006213                           ASR     (R3)            ;SET IT IN RIGHT POSITION.
013706 053713 013772                    BIS     PRIRTY,(R3)     ;GET PRIORITY IN STAT1
013712 006313                           ASL     (R3)            ;SET THEM IN RIGHT POSITION
013714 006313                           ASL     (R3)            ;  "   "   "   "   "
013716 006313                           ASL     (R3)            ;  "   "   "   "   "
013720 006313                           ASL     (R3)            ;  "   "   "   "   "
013722 053723 013766                    BIS     VECTR,(R3)+     ;GET THE VECTOR IN STAT1
013726 012223                           MOV     (R2)+,(R3)+     ;GET THE STAT2 FROM DC
013730 005723                           TST     (R3)+           ;SKIP OVER STAT3
```

```
013732  005300                          DEC     R0              ; COUNT BY 1
013734  001407                          BEQ     2$              ; ALL DONE?
013736  062737  000010  013770          ADD     #10,BASE        ; INCREMENT BASE CSR ADDRESS BY 10
013744  062737  000010  013766          ADD     #10,VECTR       ; INCREMENT VECTOR ADDRESS BY 10
013752  000747                          BR      1$              ; SET THE NEXT MAP ENTRY
013754                          2$:
013754  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
013756  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
013760  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
013762  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
013764  000207                          RTS     PC              ; RETURN
013766  000000                  VECTR:  .WORD   0
013770  000000                  BASE:   .WORD   0
013772  000000                  PRIRTY: .WORD   0
013774                          SFLT1   OUT1,INP1,4,0
013774                          SXZ


                                        ;************************** TEST 1 **************************
                                        ;* MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
                                        ;* FLOAT A 1 THROUGH REGISTER OUT1 (4)
                                        ;* FLOAT A 0THROUGH REGISTER OUT1 (4)
013774                          SXZ
                                        ;;**********************************************************

013774                          STSTN
                                ;  TEST 1
                                ;-------------
013774  012737  000001  001202  TST1:   MOV     #1,STSTNM               ; LOAD THE NO. OF THIS TEST
014002  012737  014200  001442          MOV     #TST2,NEXT              ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
014010  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
014014  014030                          MCT1
014016  104022                          ERROR   22                      ;TIME OUT ERROR...
014020  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
014024  000177  165412                  JMP     @NEXT                   ;GO TO NEXT TEST...
014030                          MCT1:
014030                          21$:
014030                                  MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
000000                                  MICPC=MICPC+1
014030  000400                          .WORD   .$.
014032                                  MOVE    BREG,SPAD (16)  ;FOR RETURN ADDRESS PURPOSES...
000001                                  MICPC=MICPC+1
014032  063236                          .WORD   .$.
014034                          SFLOT   OUT1,INP1,4,0,1,1$,2$,3$,4$
014034                          1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
000002                                  MICPC=MICPC+1
014034  000600                          .WORD   .$.
014036                          2$:
014036                                  MOVE    BREG,OUT1 (4)   ;SET THE BIT.
000003                                  MICPC=MICPC+1
014036  061224                          .WORD   .$.
014040                                  MOVE    INP1 (4),SPAD (0)       ;GET THE "FOUND" IN SCRATCH PAD.
000004                                  MICPC=MICPC+1
014040  123100                          .WORD   .$.
014042                                  MOVE    BREG,SPAD (4)
```

```
2300                000005          MICPC=MICPC+1
2301  014042       063224          .WORD   .$.
2302  014044                       $IFEQ   BREG,SPAD <0>  3$      ;CHECK THE DATA...
2303
2304
2305  014044                       SUB2C   SPAD <0>,BREG,NOP
2306                000006          MICPC=MICPC+1
2307  014044       060360          .WORD   .$.
2308  014046                       BZ      3$
2309                000007          MICPC=MICPC+1
2310  014046       101422          .WORD   .$.
2311  014050                       MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
2312                000010          MICPC=MICPC+1
2313  014050       061224          .WORD   .$.
2314  014052                       MOVE    IMP1 <4>,OUT1 <CSR5>;BAD DATA...
2315                000011          MICPC=MICPC+1
2316  014052       121105          .WORD   .$.
2317  014054                       MOVE    # 1,MEM                 ;TYPE OF ERROR...
2318                000012          MICPC=MICPC+1
2319  014054       002401          .WORD   .$.
2320  014056                       MOVE    MEM,OUT1 <CSR3>         ;
2321                000013          MICPC=MICPC+1
2322  014056       041223          .WORD   .$.
2323  014060                       MOVE    # 4,MEM         ;
2324                000014          MICPC=MICPC+1
2325  014060       002404          .WORD   .$.
2326  014062                       MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
2327                000015          MICPC=MICPC+1
2328  014062       041227          .WORD   .$.
2329  014064                       CALL    ERROR                   ;REPORT DATA ERROR.
2330  014064                       MOVE    # <MICPC+3),BREG
2331                000016          MICPC=MICPC+1
2332  014064       000420          .WORD   .$.
2333  014066                       SBR     ERROR
2334                000017          MICPC=MICPC+1
2335  014066       104400          .WORD   .$.
2336  014070                       MOVE    SPAD <4>,BREG           ;RESTORE BREG...
2337                000020          MICPC=MICPC+1
2338  014070       060604          .WORD   .$.
2339  014072                       SBR     2$                      ;LOOP ON ERROR...
2340                000021          MICPC=MICPC+1
2341  014072       100403          .WORD   .$.
2342  014074                  3$:  CALL    SCP1                    ;IS LOOP DATA SET.
2343  014074                       MOVE    # <MICPC+3),BREG
2344                000022          MICPC=MICPC+1
2345  014074       000424          .WORD   .$.
2346  014076                       SBR     SCP1
2347                000023          MICPC=MICPC+1
2348  014076       104427          .WORD   .$.
2349  014100                       MOVE    SPAD <4>,BREG
2350                000024          MICPC=MICPC+1
2351  014100       060604          .WORD   .$.
2352  014102                       SBR     2$                      ;YES, DO IT.
2353                000025          MICPC=MICPC+1
2354  014102       100403          .WORD   .$.
2355  014104                       MOVE    SPAD <4>,BREG
```

```
                    MICPC=MICPC+1
                    .WORD   .$.
                    SHFBRT                          ;NO, CONTINUE...
                    MICPC=MICPC+1
                    .WORD   .SBR!.SELB!.DBRSH
                    BBZ     4$                      ;IS IT DONE?...
                    MICPC=MICPC+1
                    .WORD   .$.
                    SBR     2$                      ;NO, CONTINUE...
                    MICPC=MICPC+1
                    .WORD   .$.
4$:
SFLOT   OUT1,INP1,4,0,0,11$,12$,13$,14$
11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
                    MICPC=MICPC+1
                    .WORD   .$.
12$:
                    MOVE    BREG,OUT1 <4>   ;SET THE BIT.
                    MICPC=MICPC+1
                    .WORD   .$.
                    MOVE    INP1 <4>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
                    MICPC=MICPC+1
                    .WORD   .$.
                    MOVE    BREG,SPAD <4>
                    MICPC=MICPC+1
                    .WORD   .$.
                    SIFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...


                    SUB2C   SPAD <0>,BREG,NOP
                    MICPC=MICPC+1
                    .WORD   .$.
                    BZ      13$
                    MICPC=MICPC+1
                    .WORD   .$.
                    MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
                    MICPC=MICPC+1
                    .WORD   .$.
                    MOVE    INP1 <4>,OUT1 <CSR5>;BAD DATA...
                    MICPC=MICPC+1
                    .WORD   .$.
                    MOVE    # 1,MEM                 ;TYPE OF ERROR...
                    MICPC=MICPC+1
                    .WORD   .$.
                    MOVE    MEM,OUT1 <CSR3>         ;
                    MICPC=MICPC+1
                    .WORD   .$.
                    MOVE    # 4,MEM         ;
                    MICPC=MICPC+1
                    .WORD   .$.
                    MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
                    MICPC=MICPC+1
                    .WORD   .$.
                    CALL    ERROR                   ;REPORT DATA ERROR.
                    MOVE    # <MICPC+3>,BREG
                    MICPC=MICPC+1
```

```
2412  014144  000450          .WORD   .$.
2413  014146                  SBR     ERROR
2414          000047          MICPC=MICPC+1
2415  014146  104400          .WORD   .$.
2416  014150                  MOVE    SPAD (4),BREG           ;RESTORE BREG...
2417          000050          MICPC=MICPC+1
2418  014150  060604          .WORD   .$.
2419  014152                  SBR     12$                     ;LOOP ON ERROR...
2420          000051          MICPC=MICPC+1
2421  014152  100433          .WORD   .$.
2422  014154          13$:    CALL    SCP1                    ;IS LOOP DATA SET.
2423  014154                  MOVE    @ (MICPC+3),BREG
2424          000052          MICPC=MICPC+1
2425  014154  000454          .WORD   .$.
2426  014156                  SBR     SCP1
2427          000053          MICPC=MICPC+1
2428  014156  104427          .WORD   .$.
2429  014160                  MOVE    SPAD (4),BREG
2430          000054          MICPC=MICPC+1
2431  014160  060604          .WORD   .$.
2432  014162                  SBR     12$                     ;YES, DO IT.
2433          000055          MICPC=MICPC+1
2434  014162  100433          .WORD   .$.
2435  014164                  MOVE    SPAD (4),BREG
2436          000056          MICPC=MICPC+1
2437  014164  060604          .WORD   .$.
2438  014166                  SHFBRT                          ;NO, CONTINUE...
2439          000057          MICPC=MICPC+1
2440  014166  061620          .WORD   .SBR!.SELB!.DBRSH
2441  014170                  BR7     12$                     ;
2442          000060          MICPC=MICPC+1
2443  014170  103433          .WORD   .$.
2444  014172          14$:
2445  014172                  CALL    SCPE
2446  014172                  MOVE    @ (MICPC+3),BREG
2447          000061          MICPC=MICPC+1
2448  014172  000463          .WORD   .$.
2449  014174                  SBR     SCPE
2450          000062          MICPC=MICPC+1
2451  014174  104454          .WORD   .$.
2452  014176                  SBR     21$
2453          000063          MICPC=MICPC+1
2454  014176  100400          .WORD   .$.
2455  014200          $FLT1   OUT1,INP1,5,0
2456  014200          $XZ
2457
2458
2459
2460
2461
2462
2463  014200          $XZ
2464
2465
2466  014200          $TSTN
2467                          ;  TEST 2
```

```
                        ;************************** TEST 2 **************************
                        ;* MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
                        ;* FLOAT A 1 THROUGH REGISTER OUT1 (5)
                        ;* FLOAT A 0THROUGH REGISTER OUT1 (5)
                        ;:*********************************************************
```

```
                                        ;--------------
014200 012737 000002 001202 TST2:   MOV     #2,$TSTNM           ; LOAD THE NO. OF THIS TEST
014206 012737 014404 001442         MOV     #TST3,NEXT          ; POINT TO THE START OF NEXT TEST.
                                                    ;R1 CONTAINS BASE KMC11 ADDRESS
014214 004737 035536                JSR     PC,LDVRWT           ;LOAD-VERIFY-WAIT.
014220 014234                       MCT2
014222 104022                       ERROR   22                  ;TIME OUT ERROR...
014224 012706 001200                MOV     #STACK,SP           ;RESET STACK...
014230 000177 165206                JMP     @NEXT               ;GO TO NEXT TEST...
014234                      MCT2:
014234                      21$:
014234                              MOVE    # 0,BREG            ;SET TO CLEAR SPAD 16
000000                              MICPC=MICPC+1
014234 000400                       .WORD   .$.
014236                              MOVE    BREG,SPAD (16)      ;FOR RETURN ADDRESS PURPOSES...
000001                              MICPC=MICPC+1
014236 063236                       .WORD   .$.
014240                      SFLOT   OUT1,INP1,5,0,1,1$,2$,3$,4$
014240                      1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
000002                              MICPC=MICPC+1
014240 000600                       .WORD   .$.
014242                      2$:
014242                              MOVE    BREG,OUT1 (5)       ;SET THE BIT.
000003                              MICPC=MICPC+1
014242 061225                       .WORD   .$.
014244                              MOVE    INP1 (5),SPAD (0)       ;GET THE "FOUND" IN SCRATCH PAD.
000004                              MICPC=MICPC+1
014244 123120                       .WORD   .$.
014246                              MOVE    BREG,SPAD (4)
000005                              MICPC=MICPC+1
014246 063224                       .WORD   .$.
014250                              $IFEQ   BREG,SPAD (0)  3$       ;CHECK THE DATA...


014250                              SUB2C   SPAD (0),BREG,NOP
000006                              MICPC=MICPC+1
014250 060360                       .WORD   .$.
014252                              BZ      3$
000007                              MICPC=MICPC+1
014252 101422                       .WORD   .$.
014254                              MOVE    BREG,OUT1 (CSR4)        ;GOOD DATA...
000010                              MICPC=MICPC+1
014254 061224                       .WORD   .$.
014256                              MOVE    INP1 (5),OUT1 (CSR5);BAD DATA...
000011                              MICPC=MICPC+1
014256 121125                       .WORD   .$.
014260                              MOVE    # 1,MEM                 ;TYPE OF ERROR...
000012                              MICPC=MICPC+1
014260 002401                       .WORD   .$.
014262                              MOVE    MEM,OUT1 (CSR3)         ;
000013                              MICPC=MICPC+1
014262 041223                       .WORD   .$.
014264                              MOVE    # 5,MEM         ;
000014                              MICPC=MICPC+1
014264 002405                       .WORD   .$.
014266                              MOVE    MEM,OUT1 (CSR7)         ;REG. ADDRESS.
```

```
                    000015                  MICPC=MICPC+1
014266  041227                              .WORD   .$.
014270                                      CALL    ERROR           ;REPORT DATA ERROR.
014270                                      MOVE    # <MICPC+3>,BREG
                    000016                  MICPC=MICPC+1
014270  000420                              .WORD   .$.
014272                                      SBR     ERROR
                    000017                  MICPC=MICPC+1
014272  104400                              .WORD   .$.
014274                                      MOVE    SPAD <4>,BREG   ;RESTORE BREG...
                    000020                  MICPC=MICPC+1
014274  060604                              .WORD   .$.
014276                                      SBR     2$              ;LOOP ON ERROR...
                    000021                  MICPC=MICPC+1
014276  100403                              .WORD   .$.
014300                              3$:     CALL    SCP1            ;IS LOOP DATA SET.
014300                                      MOVE    # <MICPC+3>,BREG
                    000022                  MICPC=MICPC+1
014300  000424                              .WORD   .$.
014302                                      SBR     SCP1
                    000023                  MICPC=MICPC+1
014302  104427                              .WORD   .$.
014304                                      MOVE    SPAD <4>,BREG
                    000024                  MICPC=MICPC+1
014304  060604                              .WORD   .$.
014306                                      SBR     2$              ;YES, DO IT.
                    000025                  MICPC=MICPC+1
014306  100403                              .WORD   .$.
014310                                      MOVE    SPAD <4>,BREG
                    000026                  MICPC=MICPC+1
014310  060604                              .WORD   .$.
014312                                      SHFBRT                  ;NO, CONTINUE...
                    000027                  MICPC=MICPC+1
014312  061620                              .WORD   .SBR!.SELB!.DBRSH
014314                                      BR7     4$              ;IS IT DONE?...
                    000030                  MICPC=MICPC+1
014314  103432                              .WORD   .$.
014316                                      SBR     2$              ;NO, CONTINUE...
                    000031                  MICPC=MICPC+1
014316  100403                              .WORD   .$.
014320                              4$:
014320                              SFLOT   OUT1,INP1,5,0,0,11$,12$,13$,14$
014320                              11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
                    000032                  MICPC=MICPC+1
014320  000577                              .WORD   .$.
014322                              12$:
014322                                      MOVE    BREG,OUT1 <5>   ;SET THE BIT.
                    000033                  MICPC=MICPC+1
014322  061225                              .WORD   .$.
014324                                      MOVE    INP1 <5>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
                    000034                  MICPC=MICPC+1
014324  123120                              .WORD   .$.
014326                                      MOVE    BREG,SPAD <4>
                    000035                  MICPC=MICPC+1
014326  063224                              .WORD   .$.
014330                                      SIFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...
```

```
2580
2581
2582 014330                    SUB2C   SPAD <0>,BREG,NOP
                 000036         MICPC=MICPC+1
2583 014330 060360              .WORD   .$.
2584 014332                     BZ      13$
                 000037         MICPC=MICPC+1
2586 014332 101452              .WORD   .$.
2587 014334                     MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
                 000040         MICPC=MICPC+1
2590 014334 061224              .WORD   .$.
2591 014336                     MOVE    INP1 <5>,OUT1 <CSR5>;BAD DATA...
                 000041         MICPC=MICPC+1
2592 014336 121125              .WORD   .$.
2593 014340                     MOVE    # 1,MEM                 ;TYPE OF ERROR...
                 000042         MICPC=MICPC+1
2594 014340 002401              .WORD   .$.
2595 014342                     MOVE    MEM,OUT1 <CSR3>         ;
                 000043         MICPC=MICPC+1
2596 014342 041223              .WORD   .$.
2597 014344                     MOVE    # 5,MEM         ;
                 000044         MICPC=MICPC+1
2600 014344 002405              .WORD   .$.
2601 014346                     MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
                 000045         MICPC=MICPC+1
2602 014346 041227              .WORD   .$.
2603 014350                     CALL    ERROR                   ;REPORT DATA ERROR.
2604 014350                     MOVE    # <MICPC+3>,BREG
                 000046         MICPC=MICPC+1
2605 014350 000450              .WORD   .$.
2606 014352                     SBR     ERROR
                 000047         MICPC=MICPC+1
2607 014352 104400              .WORD   .$.
2610 014354                     MOVE    SPAD <4>,BREG           ;RESTORE BREG...
                 000050         MICPC=MICPC+1
2611 014354 060604              .WORD   .$.
2612 014356                     SBR     12$                     ;LOOP ON ERROR...
                 000051         MICPC=MICPC+1
2613 014356 100433              .WORD   .$.
2614 014360             13$:    CALL    SCP1                    ;IS LOOP DATA SET.
2615 014360                     MOVE    # <MICPC+3>,BREG
                 000052         MICPC=MICPC+1
2616 014360 000454              .WORD   .$.
2617 014362                     SBR     SCP1
                 000053         MICPC=MICPC+1
2620 014362 104427              .WORD   .$.
2621 014364                     MOVE    SPAD <4>,BREG
                 000054         MICPC=MICPC+1
2622 014364 060604              .WORD   .$.
2623 014366                     SBR     12$                     ;YES, DO IT.
                 000055         MICPC=MICPC+1
2624 014366 100433              .WORD   .$.
2625 014370                     MOVE    SPAD <4>,BREG
                 000056         MICPC=MICPC+1
2626 014370 060604              .WORD   .$.
2627 014372                     SHFBRT                          ;NO, CONTINUE...
```

```
                    000057                          MICPC=MICPC+1
014372  061620                                      .WORD   .SBR!.SELB!.DBRSH
014374                                              BR7     145             ;
                    000060                          MICPC=MICPC+1
014374  103433                                      .WORD   .$.
014376                              145:
014376                                              CALL    SCPE
014376                                              MOVE    # <MICPC+3>,BREG
                    000061                          MICPC=MICPC+1
014376  000463                                      .WORD   .$.
014400                                              SBR     SCPE
                    000062                          MICPC=MICPC+1
014400  104454                                      .WORD   .$.
014402                                              SBR     21$
                    000063                          MICPC=MICPC+1
014402  100400                                      .WORD   .$.
014404                              $FLT1   OUT1,INP1,6,0
014404                              $XZ


                                                    ;************************** TEST 3 **************************
                                                    ;* MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
                                                    ;* FLOAT A 1 THROUGH REGISTER OUT1 <6>
                                                    ;* FLOAT A 0THROUGH REGISTER OUT1 <6>
014404                              $XZ             ;:*********************************************************************

014404                              $TSTN
                                                    ;  TEST 3
                                                    ;-------------
014404  012737  000003  001202      TST3:   MOV     #3,$TSTNM               ; LOAD THE NO. OF THIS TEST
014412  012737  014610  001442              MOV     #TST4,NEXT              ; POINT TO THE START OF NEXT TEST.
                                                                    ;R1 CONTAINS BASE KMC11 ADDRESS
014420  004737  035536                      JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
014424  014440                              MCT3
014426  104022                              ERROR   22                      ;TIME OUT ERROR...
014430  012706  001200                      MOV     #STACK,SP               ;RESET STACK...
014434  000177  165002                      JMP     @NEXT                   ;GO TO NEXT TEST...
014440                              MCT3:
014440                              21$:
014440                                              MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
                    000000                          MICPC=MICPC+1
014440  000400                                      .WORD   .$.
014442                                              MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
                    000001                          MICPC=MICPC+1
014442  063236                                      .WORD   .$.
014444                              $FLOT   OUT1,INP1,6,0,1,1$,2$,3$,4$
014444                              1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
                    000002                          MICPC=MICPC+1
014444  000600                                      .WORD   .$.
014446                              2$:
014446                                              MOVE    BREG,OUT1 <6>   ;SET THE BIT.
                    000003                          MICPC=MICPC+1
014446  061226                                      .WORD   .$.
014450                                              MOVE    INP1 <6>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
                    000004                          MICPC=MICPC+1
```

```
                                .WORD   .$
014460 123140                   MOVE    BREG,SPAD <4>
014462                          MICPC=MICPC+1
       000005                   .WORD   .$
014464 063224                   $IFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...
014466

014466                          SUB2C   SPAD <0>,BREG,NOP
       000006                   MICPC=MICPC+1
014470 060360                   .WORD   .$.
014472                          BZ      3$
       000007                   MICPC=MICPC+1
014474 101422                   .WORD   .$.
014476                          MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
       000010                   MICPC=MICPC+1
014500 061224                   .WORD   .$.
014502                          MOVE    INP1 <6>,OUT1 <CSR5>;BAD DATA...
       000011                   MICPC=MICPC+1
014504 121145                   .WORD   .$.
014506                          MOVE    # 1,MEM                 ;TYPE OF ERROR...
       000012                   MICPC=MICPC+1
014510 002401                   .WORD   .$.
014512                          MOVE    MEM,OUT1 <CSR3>         ;
       000013                   MICPC=MICPC+1
014514 041223                   .WORD   .$.
014516                          MOVE    # 6,MEM         ;
       000014                   MICPC=MICPC+1
014520 002406                   .WORD   .$.
014522                          MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
       000015                   MICPC=MICPC+1
014524 041227                   .WORD   .$.
014526                          CALL    ERROR                   ;REPORT DATA ERROR.
014530                          MOVE    # <MICPC+3>,BREG
       000016                   MICPC=MICPC+1
014532 000420                   .WORD   .$.
014534                          SBR     ERROR
       000017                   MICPC=MICPC+1
014536 104400                   .WORD   .$.
014540                          MOVE    SPAD <4>,BREG           ;RESTORE BREG...
       000020                   MICPC=MICPC+1
014542 060604                   .WORD   .$.
014544                          SBR     2$                      ;LOOP ON ERROR...
       000021                   MICPC=MICPC+1
014546 100403                   .WORD   .$.
014550                  3$:     CALL    SCP1                    ;IS LOOP DATA SET.
014552                          MOVE    # <MICPC+3>,BREG
       000022                   MICPC=MICPC+1
014554 000424                   .WORD   .$.
014556                          SBR     SCP1
       000023                   MICPC=MICPC+1
014560 104427                   .WORD   .$.
014562                          MOVE    SPAD <4>,BREG
       000024                   MICPC=MICPC+1
014564 060604                   .WORD   .$.
014566                          SBR     2$                      ;YES, DO IT.
       000025                   MICPC=MICPC+1
```

```
014512  100403                  .WORD   .$.
014514                          MOVE    SPAD (4),BREG
        000026                  MICPC=MICPC+1
014514  060604                  .WORD   .$.
014516                          SHFBRT                          ;NO, CONTINUE...
        000027                  MICPC=MICPC+1
014516  061620                  .WORD   .SBR!.SELB!.DBRSH
014520                          BBT     4$                      ;IS IT DONE?...
        000030                  MICPC=MICPC+1
014520  103432                  .WORD   .$.
014522                          SBR     2$                      ;NO, CONTINUE...
        000031                  MICPC=MICPC+1
014522  100403                  .WORD   .$.
                        4$:
                        SFLOT   OUT1,INP1,6,0,0,11$,12$,13$,14$
                        11$:    MOVE    # 177,BREG      ;START WITH BIT 7.
                                MICPC=MICPC+1
        000032                  .WORD   .$.
014524  000577
                        12$:
                                MOVE    BREG,OUT1 (6)   ;SET THE BIT.
        000033                  MICPC=MICPC+1
014526  061226                  .WORD   .$.
014530                          MOVE    INP1 (6),SPAD (0)       ;GET THE "FOUND" IN SCRATCH PAD.
        000034                  MICPC=MICPC+1
014530  123140                  .WORD   .$.
014532                          MOVE    BREG,SPAD (4)
        000035                  MICPC=MICPC+1
014532  063224                  .WORD   .$.
014534                          SIFEQ   BREG,SPAD (0)  13$      ;CHECK THE DATA...


014534
        000036                  SUB2C   SPAD (0),BREG,NOP
014534  060360                  MICPC=MICPC+1
014536                          .WORD   .$.
        000037                  BZ      13$
014536  101452                  MICPC=MICPC+1
014540                          .WORD   .$.
        000040                  MOVE    BREG,OUT1 (CSR4)        ;GOOD DATA...
014540  061224                  MICPC=MICPC+1
014542                          .WORD   .$.
        000041                  MOVE    INP1 (6),OUT1 (CSR5);BAD DATA...
014542  121145                  MICPC=MICPC+1
014544                          .WORD   .$.
        000042                  MOVE    # 1,MEM                 ;TYPE OF ERROR...
014544  002401                  MICPC=MICPC+1
014546                          .WORD   .$.
        000043                  MOVE    MEM,OUT1 (CSR3)         ;
014546  041223                  MICPC=MICPC+1
014550                          .WORD   .$.
        000044                  MOVE    # 6,MEM         ;
014550  002406                  MICPC=MICPC+1
014552                          .WORD   .$.
        000045                  MOVE    MEM,OUT1 (CSR7)         ;REG. ADDRESS.
014552  041227                  MICPC=MICPC+1
014554                          .WORD   .$.
                                CALL    EROR                    ;REPORT DATA ERROR.
```

```
014554
                000046
014554  000450
014556
                000047
014556  104400
014560
                000050
014560  060604
014562
                000051
014562  100433
014564
014564
                000052
014564  000454
014566
                000053
014566  104427
014570
                000054
014570  060604
014572
                000055
014572  100433
014574
                000056
014574  060604
014576
                000057
014576  061620
014600
                000060
014600  103433
014602
014602
014602
                000061
014602  000463
014604
                000062
014604  104454
014606
                000063
  4606  100400
  4610
  4610
014610
```

```
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     ERROR
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,BREG           ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     13$                     ;LOOP ON ERROR...
                MICPC=MICPC+1
                .WORD   .$.
13$:            CALL    SCP1                    ;IS LOOP DATA SET.
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     SCP1
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     14$                     ;YES, DO IT.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SHFBRT                          ;NO, CONTINUE...
                MICPC=MICPC+1
                .WORD   .SBR!.SELB!.DBRSH
                BR7     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
                .WORD   .$.
$FLT1   OUT1,INP1,7,0
$XZ


                ;*************************** TEST 4 ****************************
                ;* MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
                ;* FLOAT A 1 THROUGH REGISTER OUT1 <7>
                ;* FLOAT A 0THROUGH REGISTER OUT1 <7>
$XZ
                ;:***************************************************************
```

```
014610                                  $TSTN
                                                ;  TEST 4
                                                ;-------------
014610  012737  000004  001202  TST4:   MOV     #4,$TSTNM               ; LOAD THE NO. OF THIS TEST
014616  012737  015014  001442          MOV     #TST5,NEXT              ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
014624  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
014630  014444                          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                          21$:
014644                                  MOVE    # 0,BREG         ;SET TO CLEAR SPAD 16
000000                                  MICPC=MICPC+1
014646  000400                          .WORD   .$.
014646                                  MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
000001                                  MICPC=MICPC+1
014650  063236                          .WORD   .$.
014650                          SFLOT   OUT1,INP1,7,0,1,1$,2$,3$,4$
014650                          1$:     MOVE    # 200,BREG      ;START WITH BIT 7.
000002                                  MICPC=MICPC+1
014650  000600                          .WORD   .$.
014652                          2$:
014652                                  MOVE    BREG,OUT1 <7>   ;SET THE BIT.
000003                                  MICPC=MICPC+1
014652  061227                          .WORD   .$.
014654                                  MOVE    INP1 <7>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
000004                                  MICPC=MICPC+1
014654  123160                          .WORD   .$.
014656                                  MOVE    BREG,SPAD <4>
000005                                  MICPC=MICPC+1
014656  063224                          .WORD   .$.
014660                                  $IFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...


014660                                  SUB2C   SPAD <0>,BREG,NOP
000006                                  MICPC=MICPC+1
014660  060360                          .WORD   .$.
014662                                  BZ      3$
000007                                  MICPC=MICPC+1
014662  101422                          .WORD   .$.
014664                                  MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
000010                                  MICPC=MICPC+1
014664  061224                          .WORD   .$.
014666                                  MOVE    INP1 <7>,OUT1 <CSR5>;BAD DATA...
000011                                  MICPC=MICPC+1
014666  121165                          .WORD   .$.
014670                                  MOVE    # 1,MEM                 ;TYPE OF ERROR...
000012                                  MICPC=MICPC+1
014670  002401                          .WORD   .$.
014672                                  MOVE    MEM,OUT1 <CSR3>         ;
000013                                  MICPC=MICPC+1
014672  041223                          .WORD   .$.
014674                                  MOVE    # 7,MEM         ;
000014                                  MICPC=MICPC+1
```

```
2916  014674  002407                .WORD   .$.
2917  014676                        MOVE    MEM,OUT1 (CSR7)         ;REG. ADDRESS.
2918          000015                MICPC=MICPC+1
2919  014676  041227                .WORD   .$.
2920  014700                        CALL    ERROR                   ;REPORT DATA ERROR.
2921  014700                        MOVE    # (MICPC+3),BREG
2922          000016                MICPC=MICPC+1
2923  014700  000420                .WORD   .$.
2924  014702                        SBR     ERROR
2925          000017                MICPC=MICPC+1
2926  014702  104400                .WORD   .$.
2927  014704                        MOVE    SPAD (4),BREG           ;RESTORE BREG...
2928          000020                MICPC=MICPC+1
2929  014704  060604                .WORD   .$.
2930  014706                        SBR     2$                      ;LOOP ON ERROR...
2931          000021                MICPC=MICPC+1
2932  014706  100403                .WORD   .$.
2933  014710                3$:     CALL    SCP1                    ;IS LOOP DATA SET.
2934  014710                        MOVE    # (MICPC+3),BREG
2935          000022                MICPC=MICPC+1
2936  014710  000424                .WORD   .$.
2937  014712                        SBR     SCP1
2938          000023                MICPC=MICPC+1
2939  014712  104427                .WORD   .$.
2940  014714                        MOVE    SPAD (4),BREG
2941          000024                MICPC=MICPC+1
2942  014714  060604                .WORD   .$.
2943  014716                        SBR     2$                      ;YES, DO IT.
2944          000025                MICPC=MICPC+1
2945  014716  100403                .WORD   .$.
2946  014720                        MOVE    SPAD (4),BREG
2947          000026                MICPC=MICPC+1
2948  014720  060604                .WORD   .$.
2949  014722                        SHFBRT                          ;NO, CONTINUE...
2950          000027                MICPC=MICPC+1
2951  014722  061620                .WORD   .SBR!.SELB!.DBRSH
2952  014724                        BB7     4$                      ;IS IT DONE?...
2953          000030                MICPC=MICPC+1
2954  014724  103432                .WORD   .$.
2955  014726                        SBR     2$                      ;NO, CONTINUE...
2956          000031                MICPC=MICPC+1
2957  014726  100403                .WORD   .$.
2958  014730                4$:
2959  014730                SFLOT   OUT1,INP1,7,0,0,11$,12$,13$,14$
2960  014730                11$:    MOVE  # 177,BREG  ;START WITH BIT 7.
2961          000032                MICPC=MICPC+1
2962  014730  000577                .WORD   .$.
2963  014732                12$:
2964  014732                        MOVE    BREG,OUT1 (7)   ;SET THE BIT.
2965          000033                MICPC=MICPC+1
2966  014732  061227                .WORD   .$.
2967  014734                        MOVE    INP1 (7),SPAD (0)       ;GET THE "FOUND" IN SCRATCH PAD.
2968          000034                MICPC=MICPC+1
2969  014734  123160                .WORD   .$.
2970  014736                        MOVE    BREG,SPAD (4)
2971          000035                MICPC=MICPC+1
```

```
                                                .WORD   .$
014736 063224                                   SIFEQ   BREG,SPAD <0>  135      ;CHECK THE DATA...
014740
014740                                          SUB2C   SPAD <0>,BREG,NOP
                000026                          MICPC=MICPC+1
014740 060360                                   .WORD   .$
014742                                          BNZ     135
                000027                          MICPC=MICPC+1
014742 101462                                   .WORD   .$
014744                                          MOVE    BREG,OUT1 <CSR4>         ;GOOD DATA...
                000040                          MICPC=MICPC+1
014744 061224                                   .WORD   .$
014746                                          MOVE    INP1 <7>,OUT1 <CSR5>;BAD DATA...
                000041                          MICPC=MICPC+1
014746 121165                                   .WORD   .$
014750                                          MOVE    # 1,MEM                  ;TYPE OF ERROR...
                000042                          MICPC=MICPC+1
014750 002401                                   .WORD   .$
014752                                          MOVE    MEM,OUT1 <CSR3>          ;
                000043                          MICPC=MICPC+1
014752 041223                                   .WORD   .$
014754                                          MOVE    # 7,MEM          ;
                000044                          MICPC=MICPC+1
014754 002407                                   .WORD   .$
014756                                          MOVE    MEM,OUT1 <CSR7>          ;REG. ADDRESS.
                000045                          MICPC=MICPC+1
014756 041227                                   .WORD   .$
014760                                          CALL    ERROR                    ;REPORT DATA ERROR.
014760                                          MOVE    # <MICPC+3>,BREG
                000046                          MICPC=MICPC+1
014760 000450                                   .WORD   .$
014762                                          SBR     ERROR
                000047                          MICPC=MICPC+1
014762 104400                                   .WORD   .$
014764                                          MOVE    SPAD <4>,BREG            ;RESTORE BREG...
                000050                          MICPC=MICPC+1
014764 060604                                   .WORD   .$
014766                                          SBR     125                      ;LOOP ON ERROR...
                000051                          MICPC=MICPC+1
014766 100433                                   .WORD   .$
014770                                  135:    CALL    SCP1                     ;IS LOOP DATA SET.
014770                                          MOVE    # <MICPC+3>,BREG
                000052                          MICPC=MICPC+1
014770 000454                                   .WORD   .$
014772                                          SBR     SCP1
                000053                          MICPC=MICPC+1
014772 104427                                   .WORD   .$
014774                                          MOVE    SPAD <4>,BREG
                000054                          MICPC=MICPC+1
014774 060604                                   .WORD   .$
014776                                          SBR     125                      ;YES, DO IT.
                000055                          MICPC=MICPC+1
014776 100433                                   .WORD   .$
015000                                          MOVE    SPAD <4>,BREG
                000056                          MICPC=MICPC+1
```

```
015000  060604                  .WORD   .$.
015002                          SHFBRT                          ;NO, CONTINUE...
        000057                  MICPC=MICPC+1
015002  061620                  .WORD   .SBR!.SELB!.DBRSH
015004                          BR7     12$             ;
        000060                  MICPC=MICPC+1
015004  103433                  .WORD   .$.
015006                  14$:
015006                          CALL    SCPE
015006                          MOVE    # <MICPC+3>,BREG
        000061                  MICPC=MICPC+1
015006  000463                  .WORD   .$.
015010                          SBR     SCPE
        000062                  MICPC=MICPC+1
015010  104454                  .WORD   .$.
015012                          SBR     21$
        000063                  MICPC=MICPC+1
015012  100400                  .WORD   .$.
015014                  $FLT1   OUT1,INP1,10,236
015014                  $XZ


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

015014                  $TSTN
                                ;  TEST 5
                                ;-------------
015014  012737  000005  001202  TST5:   MOV     #5,$TSTNM               ; LOAD THE NO. OF THIS TEST
015022  012737  015234  001442          MOV     #TST6,NEXT              ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
015030  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
015034  015050                          MCT5
015036  104022                          ERROR   22                      ;TIME OUT ERROR...
015040  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
015044  000177  164372                  JMP     @NEXT                   ;GO TO NEXT TEST...
015050                  MCT5:
015050                  21$:
015050                          MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
        000000                  MICPC=MICPC+1
015050  000400                  .WORD   .$.
015052                          MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
        000001                  MICPC=MICPC+1
015052  063236                  .WORD   .$.
015054                  $FLOT   OUT1,INP1,10,236,1,1$,2$,3$,4$
015054                  1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
        000002                  MICPC=MICPC+1
015054  000600                  .WORD   .$.
015056                  2$:
015056                          MOVE    BREG,SPAD <4>           ;SAVE BREGISTER.
        000003                  MICPC=MICPC+1
```

```
3084  015056  063224                .WORD   .$.
3085  015060                        MOVE    # 236,MEM               ;CLEAR THE UNWANTED BITS
3086          000004                MICPC=MICPC+1
3087  015060  003636                .WORD   .$.
3088  015062                        MOVE    MEM,SPAD <1>            ;CLEAR THE UNWANTED BITS.
3089          000005                MICPC=MICPC+1
3090  015062  043221                .WORD   .$.
3091  015064                        AND     SPAD <1>,BREG,BREG      ;CLEAR THE UNWANTED BITS.
3092          000006                MICPC=MICPC+1
3093  015064  060661                .WORD   .$.
3094  015066                        MOVE    BREG,OUT1 <10>  ;SET THE BIT.
3095          000007                MICPC=MICPC+1
3096  015066  061230                .WORD   .$.
3097  015070                        MOVE    INP1 <10>,SPAD <0>      ;GET THE "FOUND" IN SCRATCH PAD.
3098          000010                MICPC=MICPC+1
3099  015070  123200                .WORD   .$.
3100  015072                        $IFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...
3101
3102
3103  015072                        SUB2C   SPAD <0>,BREG,NCP
3104          000011                MICPC=MICPC+1
3105  015072  060360                .WORD   .$.
3106  015074                        BZ      3$
3107          000012                MICPC=MICPC+1
3108  015074  101425                .WORD   .$.
3109  015076                        MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
3110          000013                MICPC=MICPC+1
3111  015076  061224                .WORD   .$.
3112  015100                        MOVE    INP1 <10>,OUT1 <CSR5>;BAD DATA...
3113          000014                MICPC=MICPC+1
3114  015100  121205                .WORD   .$.
3115  015102                        MOVE    # 1,MEM                 ;TYPE OF ERROR...
3116          000015                MICPC=MICPC+1
3117  015102  002401                .WORD   .$.
3118  015104                        MOVE    MEM,OUT1 <CSR3>         ;
3119          000016                MICPC=MICPC+1
3120  015104  041223                .WORD   .$.
3121  015106                        MOVE    # 10,MEM                ;
3122          000017                MICPC=MICPC+1
3123  015106  002410                .WORD   .$.
3124  015110                        MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
3125          000020                MICPC=MICPC+1
3126  015110  041227                .WORD   .$.
3127  015112                        CALL    ERROR                   ;REPORT DATA ERROR.
3128  015112                        MOVE    # <MICPC+3),BREG
3129          000021                MICPC=MICPC+1
3130  015112  000423                .WORD   .$.
3131  015114                        $BR     ERROR
3132          000022                MICPC=MICPC+1
3133  015114  104400                .WORD   .$.
3134  015116                        MOVE    SPAD <4),BREG           ;RESTORE BREG...
3135          000023                MICPC=MICPC+1
3136  015116  060604                .WORD   .$.
3137  015120                        $BR     2$                      ;LOOP ON ERROR...
3138          000024                MICPC=MICPC+1
3139  015120  100403                .WORD   .$.
```

```
3140  015122                        3S:     CALL    SCP1                    ;IS LOOP DATA SET.
3141  015122                                MOVE    # <MICPC+3>,BREG
3142          000025                        MICPC=MICPC+1
3143  015122  000427                        .WORD   .$.
3144  015124                                SBR     SCP1
3145          000026                        MICPC=MICPC+1
3146  015124  104427                        .WORD   .$.
3147  015126                                MOVE    SPAD <4>,BREG
3148          000027                        MICPC=MICPC+1
3149  015126  060604                        .WORD   .$.
3150  015130                                SBR     2S                      ;YES, DO IT.
3152          000030                        MICPC=MICPC+1
3152  015130  100403                        .WORD   .$.
3153  015132                                MOVE    SPAD <4>,BREG
3154          000031                        MICPC=MICPC+1
3155  015132  060604                        .WORD   .$.
3156  015134                                SHFBRT                          ;NO, CONTINUE...
3157          000032                        MICPC=MICPC+1
3158  015134  061620                        .WORD   .SBR!.SELB!.DBRSH
3159  015136                                BBZ     4S                      ;IS IT DONE?...
3160          000033                        MICPC=MICPC+1
3161  015136  103435                        .WORD   .$.
3162  015140                                SBR     2S                      ;NO, CONTINUE...
3163          000034                        MICPC=MICPC+1
3164  015140  100403                        .WORD   .$.
3165  015142                        4S:
3166  015142                        $FLOT   OUT1,INP1,10,236,0,11$,12$,13$,14$
3167  015142                        11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
3168          000035                        MICPC=MICPC+1
3169  015142  000577                        .WORD   .$.
3170  015144                        12$:
3171  015144                                MOVE    BREG,SPAD <4>           ;SAVE BREGISTER.
3172          000036                        MICPC=MICPC+1
3173  015144  063224                        .WORD   .$.
3174  015146                                MOVE    # 236,MEM               ;CLEAR THE UNWANTED BITS
3175          000037                        MICPC=MICPC+1
3176  015146  002636                        .WORD   .$.
3177  015150                                MOVE    MEM,SPAD <1>            ;CLEAR THE UNWANTED BITS.
3178          000040                        MICPC=MICPC+1
3179  015150  043221                        .WORD   .$.
3180  015152                                AND     SPAD <1>,BREG,BREG      ;CLEAR THE UNWANTED BITS.
3181          000041                        MICPC=MICPC+1
3182  015152  060661                        .WORD   .$.
3183  015154                                MOVE    BREG,OUT1 <10>  ;SET THE BIT.
3184          000042                        MICPC=MICPC+1
3185  015154  061230                        .WORD   .$.
3186  015156                                MOVE    INP1 <10>,SPAD <0>      ;GET THE "FOUND" IN SCRATCH PAD.
3187          000043                        MICPC=MICPC+1
3188  015156  123200                        .WORD   .$.
3189  015160                                $IFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...
3190
3191
3192  015160                                SUB2C   SPAD <0>,BREG,NOP
3193          000044                        MICPC=MICPC+1
3194  015160  060360                        .WORD   .$.
3195  015162                                BZ      13$
```

```
3196                    000045
3197  015162  101460
3198  015164
3199                    000046
3200  015164  061224
3201  015166
3202                    000047
3203  015166  121205
3204  015170
3205                    000050
3206  015170  002401
3207  015172
3208                    000051
3209  015172  041223
3210  015174
3211                    000052
3212  015174  002410
3213  015176
3214                    000053
3215  015176  041227
3216  015200
3217  015200
3218                    000054
3219  015200  000456
3220  015202
3221                    000055
3222  015202  104400
3223  015204
3224                    000056
3225  015204  060604
3226  015206
3227                    000057
3228  015206  100436
3229  015210
3230  015210
3231                    000060
3232  015210  000462
3233  015212
3234                    000061
3235  015212  104427
3236  015214
3237                    000062
3238  015214  060604
3239  015216
3240                    000063
3241  015216  100436
3242  015220
3243                    000064
3244  015220  060604
3245  015222
3246                    000065
3247  015222  061620
3248  015224
3249                    000066
3250  015224  103436
3251  015226
```

```
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    INP1 <10>,OUT1 <CSR5>;BAD DATA...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 1,MEM                 ;TYPE OF ERROR...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,OUT1 <CSR3>         ;
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 10,MEM                ;
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
        MICPC=MICPC+1
        .WORD   .$.
        CALL    EROR                    ;REPORT DATA ERROR.
        MOVE    # <MICPC+3>,BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     EROR
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD <4>,BREG           ;RESTORE BREG...
        MICPC=MICPC+1
        .WORD   .$.
        SBR     12$                     ;LOOP ON ERROR...
        MICPC=MICPC+1
        .WORD   .$.
13$:    CALL    SCP1                    ;IS LOOP DATA SET.
        MOVE    # <MICPC+3>,BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     SCP1
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD <4>,BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     12$                     ;YES, DO IT.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD <4>,BREG
        MICPC=MICPC+1
        .WORD   .$.
        SHFBRT                          ;NO, CONTINUE...
        MICPC=MICPC+1
        .WORD   .SBR!.SELB!.DBRSH
        BB7     12$             ;
        MICPC=MICPC+1
        .WORD   .$.
14$:
```

```
015226                          CALL    SCPE
015226                          MOVE    @#<MICPC+3>,BREG
                                MICPC=MICPC+1
000067                          .WORD   .$.
015226  000471                  SBR     SCPE
015230                          MICPC=MICPC+1
000070                          .WORD   .$.
015230  104454                  SBR     21$
015232                          MICPC=MICPC+1
000071                          .WORD   .$.
015232  100400          $FLT1   OUT1,INPI,I1,015
015234          $XZ
015234          $XZ

                        ;************************** TEST 6 **************************
                        ;* MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
                        ;* FLOAT A 1 THROUGH REGISTER OUT1 <I1>
                        ;* FLOAT A 0THROUGH REGISTER OUT1 <I1>
                        ;* THE BR RQ BIT, PGM CLOCK BIT, FORCE POWER FAIL BIT
                        ;* (BITS 7,4,1) ARE ALL MASKED DURING THIS TEST
015234          $XZ
                        ;:*******************************************************************

015234          $TSTN
                        ;  TEST 6
                        ;--------------
015234  012737  000006  001202  TST6:   MOV     #6,$TSTNM               ; LOAD THE NO. OF THIS TEST
015242  012737  015470  001442          MOV     #TST7,NEXT              ; POINT TO THE START OF NEXT TEST.
                                ;R1 CONTAINS BASE KMC11 ADDRESS
015250  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
015254  015270
015256  104022                          MCT6
015260  012706  001200                  ERROR   22                      ;TIME OUT ERROR...
015264  000177  164152                  MOV     #STACK,SP               ;RESET STACK...
                                        JMP     @NEXT                   ;GO TO NEXT TEST...
015270                          MCT6:
015270                          21$:
015270                                  MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
000000                                  MICPC=MICPC+1
015270  000400                          .WORD   .$.
015272                                  MOVE    BREG,SPAD (16)  ;FOR RETURN ADDRESS PURPOSES...
000001                                  MICPC=MICPC+1
015272  063236                          .WORD   .$.
015274                          SFLOT   OUT1,INPI,I1,015,1,1$,2$,3$,4$
015274                          1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
000002                                  MICPC=MICPC+1
015274  000600                          .WORD   .$.
015276                          2$:
015276                                  MOVE    BREG,SPAD (4)           ;SAVE BREGISTER.
000003                                  MICPC=MICPC+1
015276  063224                          .WORD   .$.
015300                                  MOVE    # 015,MEM               ;CLEAR THE UNWANTED BITS
000004                                  MICPC=MICPC+1
015300  002415                          .WORD   .$.
015302                                  MOVE    MEM,SPAD (1)            ;CLEAR THE UNWANTED BITS.
000005                                  MICPC=MICPC+1
015302  043221                          .WORD   .$.
```

```
3308  015304
3309          000006
3310  015304  060661      AND     SPAD (1),BREG,BREG      ;CLEAR THE UNWANTED BITS.
                          MICPC=MICPC+1
                          .WORD   .$.
3311  015306              MOVE    BREG,OUT1 (11)  ;SET THE BIT.
3312          000007      MICPC=MICPC+1
3313  015306  061231      .WORD   .$.
3314  015310              MOVE    INP1 (11),SPAD (0)      ;GET THE "FOUND" IN SCRATCH PAD.
3315          000010      MICPC=MICPC+1
3316  015310  123220      .WORD   .$.
3317  015312              MOVE    # 20,MEM                ;SET TO SET PGM.CLK BIT.
3318          000011      MICPC=MICPC+1
3319  015312  002420      .WORD   .$.
3320  015314              MOVE    MEM,SPAD  (1)           ;  "   "   "   "   "   "
3321          000012      MICPC=MICPC+1
3322  015314  043221      .WORD   .$.
3323  015316              OR      SPAD (1),BREG,BREG      ;SET PGM.CLK. BIT IN BREG.
3324          000013      MICPC=MICPC+1
3325  015316  060701      .WORD   .$.
3326  015320              SIFEQ   BREG,SPAD (0)  3$       ;CHECK THE DATA...
3327
3328  015320
3329          000014      SUB2C   SPAD (0),BREG,NOP
3330  015320  060360      MICPC=MICPC+1
3331  015322              .WORD   .$.
3332          000015      BZ      3$
3333  015322  101430      MICPC=MICPC+1
3334  015324              .WORD   .$.
3335          000016      MOVE    BREG,OUT1 (CSR4)        ;GOOD DATA...
3336  015324  061224      MICPC=MICPC+1
3337  015326              .WORD   .$.
3338          000017      MOVE    INP1 (11),OUT1 (CSR5);BAD DATA...
3339  015326  121225      MICPC=MICPC+1
3340  015330              .WORD   .$.
3341          000020      MOVE    # 1,MEM                 ;TYPE OF ERROR...
3342  015330  002401      MICPC=MICPC+1
3343  015332              .WORD   .$.
3344          000021      MOVE    MEM,OUT1 (CSR3)         ;
3345  015332  041223      MICPC=MICPC+1
3346  015334              .WORD   .$.
3347          000022      MOVE    # 11,MEM                ;
3348  015334  002411      MICPC=MICPC+1
3349  015336              .WORD   .$.
3350          000023      MOVE    MEM,OUT1 (CSR7)         ;REG. ADDRESS.
3351  015336  041227      MICPC=MICPC+1
3352  015340              .WORD   .$.
3353  015340              CALL    EROR                    ;REPORT DATA ERROR.
3354          000024      MOVE    # (MICPC+3),BREG
3355  015340  000426      MICPC=MICPC+1
3356  015342              .WORD   .$.
3357          000025      SBR     EROR
3358  015342  104400      MICPC=MICPC+1
3359  015344              .WORD   .$.
3360          000026      MOVE    SPAD (4),BREG           ;RESTORE BREG...
3361  015344  060604      MICPC=MICPC+1
3362  015346              .WORD   .$.
3363                      SBR     2$                      ;LOOP ON ERROR...
```

```
3364  015346  000027
3365  015346  100403
3366  015350
3367  015350
3368  015352  000030
3369  015352  000432
3370  015354
3371  015354  000031
3372  015356  104427
3373  015360
3374  015360  000032
3375  015362  060604
3376  015364
3377  015364  000033
3400  015366  100403
3401  015370
3402  015370  000034
3403  015372  060604
3404  015374
3405  015374  000035
3406  015376  061620
3407  016000
3410  016000  000036
3411  016002  103440
3412  016004
3413  016004  000037
3414  016006  100403
3415  016010
3416  016010
3417  016012  000040
3418  016012  000577
3419  016014
3420  016014
3421  016016  000041
3422  016016  063224
3423  016020
3424  016020  000042
3425  016022  002415
3426  016024
3427  016024  000043
3430  016026  043221
3431  016030
3432  016030  000044
3433  016032  060661
3434  016034
3435  016034  000045
3436  016036  061231
3437  016040
3440  016040  000046
3441  016042  123220
3442  016044
3443  016044  000047
3444  016046  002420
3445  016050
3446  016050  000050
```

```
                MICPC=MICPC+1
                .WORD   $.
35:             CALL    SCP1                    ;IS LOOP DATA SET.
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     SCP1
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     2$                      ;YES, DO IT.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SHFBRT                          ;NO, CONTINUE...
                MICPC=MICPC+1
                .WORD   .SBR!.SELB!.DBRSH
                BBZ     4$                      ;IS IT DONE?...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     2$                      ;NO, CONTINUE...
                MICPC=MICPC+1
                .WORD   .$.
4$:
SFLOT   OUT1,INP1,11,015,0,11$,12$,13$,14$
11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
                MICPC=MICPC+1
                .WORD   .$.
12$:
                MOVE    BREG,SPAD <4>           ;SAVE BREGISTER.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 015,MEM               ;CLEAR THE UNWANTED BITS
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,SPAD <1>            ;CLEAR THE UNWANTED BITS.  -
                MICPC=MICPC+1
                .WORD   .$.
                AND     SPAD <1>,BREG,BREG      ;CLEAR THE UNWANTED BITS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT1 <11>  ;SET THE BIT.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    INP1 <11>,SPAD <0>      ;GET THE "FOUND" IN SCRATCH PAD.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 20,MEM                ;SET TO SET PGM.CLK BIT.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,SPAD  <1>           ; "  "  "  "  "  "  .
                MICPC=MICPC+1
```

```
3420 015410 043221                      .WORD   .$.
3421 015412                             OR      SPAD <1>,BREG,BREG      ;SET PGM.CLK. BIT IN BREG.
                 000051                 MICPC=MICPC+1
3423 015412 060701                      .WORD   .$.
3424 015414                             SIFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...

3427 015414                             SUB2C   SPAD <0>,BREG,NOP
                 000052                 MICPC=MICPC+1
     015414 060360                      .WORD   .$.
     015416                             BZ      13$
                 000053                 MICPC=MICPC+1
     015416 101466                      .WORD   .$.
     015420                             MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
                 000054                 MICPC=MICPC+1
     015420 061224                      .WORD   .$.
     015422                             MOVE    INP1 <11>,OUT1 <CSR5>;BAD DATA...
                 000055                 MICPC=MICPC+1
     015422 121225                      .WORD   .$.
     015424                             MOVE    # 1,MEM                 ;TYPE OF ERROR...
                 000056                 MICPC=MICPC+1
3441 015424 002401                      .WORD   .$.
     015426                             MOVE    MEM,OUT1 <CSR3>         ;
                 000057                 MICPC=MICPC+1
3444 015426 041223                      .WORD   .$.
3445 015430                             MOVE    # 11,MEM                ;
                 000060                 MICPC=MICPC+1
3447 015430 002411                      .WORD   .$.
     015432                             MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
                 000061                 MICPC=MICPC+1
3450 015432 041227                      .WORD   .$.
     015434                             CALL    ERROR                   ;REPORT DATA ERROR.
     015434                             MOVE    # <MICPC+3>,BREG
                 000062                 MICPC=MICPC+1
     015434 000464                      .WORD   .$.
     015436                             SBR     ERROR
                 000063                 MICPC=MICPC+1
     015436 104400                      .WORD   .$.
     015440                             MOVE    SPAD <4>,BREG           ;RESTORE BREG...
                 000064                 MICPC=MICPC+1
     015440 060604                      .WORD   .$.
     015442                             SBR     12$                     ;LOOP ON ERROR...
                 000065                 MICPC=MICPC+1
     015442 100441                      .WORD   .$.
     015444                     13$:    CALL    SCP1                    ;IS LOOP DATA SET.
     015444                             MOVE    # <MICPC+3>,BREG
                 000066                 MICPC=MICPC+1
     015444 000470                      .WORD   .$.
     015446                             SBR     SCP1
                 000067                 MICPC=MICPC+1
3470 015446 104427                      .WORD   .$.
3471 015450                             MOVE    SPAD <4>,BREG
                 000070                 MICPC=MICPC+1
3473 015450 060604                      .WORD   .$.
3474 015452                             SBR     12$                     ;YES, DO IT.
3475        000071                      MICPC=MICPC+1
```

```
015462  100441                          .WORD   .$
015464                                  MOVE    SPAD (4),BREG
                                        MICPC=MICPC+1
015466  000072                          .WORD   .$.
015470  060604                          SUBRT
015472                                  MICPC=MICPC+1                   ;NO, CONTINUE...
                                        .WORD   .SBR!.SELB!.DBRSH
015474  000073                          BR7     14$                     ;
015476  061620                          MICPC=MICPC+1
015500                                  .WORD   .$.
015502  000074                  14$:
015504  103441                          CALL    SCPE
015506                                  MOVE    # (MICPC+3),BREG
015510                                  MICPC=MICPC+1
015512  000075                          .WORD   .$.
015514  000477                          SBR     SCPE
015516                                  MICPC=MICPC+1
015520  000076                          .WORD   .$.
015522  104454                          SBR     21$
015524                                  MICPC=MICPC+1
015526  000077                          .WORD   .$.
015530  100400                  $FLT1   OUTO,INPO,0,0
015532                          $XZ


                                        ;************************** TEST 7 **************************
                                        ;* MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
                                        ;* FLOAT A 1 THROUGH REGISTER OUTO (0)
                                        ;* FLOAT A 0THROUGH REGISTER OUTO (0)
015534                          $XZ
                                        ;:*****************************************************************

015536                          $TSTN
                                        ;  TEST 7
                                        ;-------------

015540  012737  000007  001202  TST7:   MOV     #7,$TSTNM               ; LOAD THE NO. OF THIS TEST
015546  012737  015674  001442          MOV     #TST10,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
015554  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
015560  015624                          MCT7
015562  104022                          ERROR   22                      ;TIME OUT ERROR...
015564  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
015570  000177  163716                  JMP     @NEXT                   ;GO TO NEXT TEST...
015574                          MCT7:
015576                          21$:
015600  000000                          MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
015602  015624  000400                  MICPC=MICPC+1
015604  015626                          .WORD   .$.
                                        MOVE    BREG,SPAD (16)  ;FOR RETURN ADDRESS PURPOSES...
015610  000001                          MICPC=MICPC+1
015612  015626  063236                  .WORD   .$.
015614  015630                  $FLOT   OUTO,INPO,0,0,1,1$,2$,3$,4$
015616  015630                  1$:  MOVE   #'200,BREG  ;START WITH BIT 7.
                                        MICPC=MICPC+1
015620  000002                          .WORD   .$.
015622  015630  000600          2$:
015624  015632
```

```
015532                          MOVE    BREG,OUT0 <0>   ;SET THE BIT.
                 000003         MICPC=MICPC+1
015532 062220                   .WORD   .$.
015534                          MOVE    INP0 <0>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
                 000004         MICPC=MICPC+1
015534 023000                   .WORD   .$.
015536                          MOVE    BREG,SPAD <4>
                 000005         MICPC=MICPC+1
015536 063224                   .WORD   .$.
015540                          $IFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...

015540                          SUB2C   SPAD <0>,BREG,NOP
                 000006         MICPC=MICPC+1
015540 060360                   .WORD   .$.
015542                          BZ      3$
                 000007         MICPC=MICPC+1
015542 101422                   .WORD   .$.
015544                          MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
                 000010         MICPC=MICPC+1
015544 061224                   .WORD   .$.
015546                          MOVE    INP0 <0>,OUT1 <CSR5>;BAD DATA...
                 000011         MICPC=MICPC+1
015546 021005                   .WORD   .$.
015550                          MOVE    # 1,MEM                 ;TYPE OF ERROR...
                 000012         MICPC=MICPC+1
015550 002401                   .WORD   .$.
015552                          MOVE    MEM,OUT1 <CSR3>         ;
                 000013         MICPC=MICPC+1
015552 041223                   .WORD   .$.
015554                          MOVE    # 0,MEM         ;
                 000014         MICPC=MICPC+1
015554 002400                   .WORD   .$.
015556                          MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
                 000015         MICPC=MICPC+1
015556 041227                   .WORD   .$.
015560                          CALL    ERROR                   ;REPORT DATA ERROR.
015560                          MOVE    # <MICPC+3>,BREG
                 000016         MICPC=MICPC+1
015560 000420                   .WORD   .$.
015562                          SBR     ERROR
                 000017         MICPC=MICPC+1
015562 104400                   .WORD   .$.
015564                          MOVE    SPAD <4>,BREG           ;RESTORE BREG...
                 000020         MICPC=MICPC+1
015564 060604                   .WORD   .$.
015566                          SBR     2$                      ;LOOP ON ERROR...
                 000021         MICPC=MICPC+1
015566 100403                   .WORD   .$.
015570                  3$:     CALL    SCP1                    ;IS LOOP DATA SET.
015570                          MOVE    # <MICPC+3>,BREG
                 000022         MICPC=MICPC+1
015570 000424                   .WORD   .$.
015572                          SBR     SCP1
                 000023         MICPC=MICPC+1
015572 104427                   .WORD   .$.
```

```
3588  015574                    MOVE    SPAD <4>,BREG
3589                  000024    MICPC=MICPC+1
3590  015574  060604            .WORD   .$.
3591  015576                    SBR     2$                      ;YES, DO IT.
3592                  000025    MICPC=MICPC+1
3593  015576  100403            .WORD   .$.
3594  015600                    MOVE    SPAD <4>,BREG
3595                  000026    MICPC=MICPC+1
3596  015600  060604            .WORD   .$.
3597  015602                    SHFBRT                          ;NO, CONTINUE...
3598                  000027    MICPC=MICPC+1
3599  015602  06'620            .WORD   .SBR!.SELB!.DBRSH
3600  015604                    BR7     4$                      ;IS IT DONE?...
3601                  000030    MICPC=MICPC+1
3602  015604  103432            .WORD   .$.
3603  015606                    SBR     2$                      ;NO, CONTINUE...
3604                  000031    MICPC=MICPC+1
3605  015606  100403            .WORD   .$.
3606  015610              4$:
3607  015610              SFLOT   OUT0,INP0,0,0,0,11$,12$,13$,14$
3608  015610              11$:  MOVE    # 177,BREG  ;START WITH BIT 7.
3609                  000032    MICPC=MICPC+1
3610  015610  000577            .WORD   .$.
3611  015612              1$:
3612  015612                    MOVE    BREG,OUT0 <0>   ;SET THE BIT.
3613                  000033    MICPC=MICPC+1
3614  015612  062220            .WORD   .$.
3615  015614                    MOVE    INP0 <0>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
3616                  000034    MICPC=MICPC+1
3617  015614  023000            .WORD   .$.
3618  015616                    MOVE    BREG,SPAD <4>
3619                  000035    MICPC=MICPC+1
3620  015616  063224            .WORD   .$.
3621  015620                    $IFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...

3624  015620              SUB2C   SPAD <0>,BREG,NOP
3625                  000036    MICPC=MICPC+1
3626  015620  060360            .WORD   .$.
3627  015622                    BZ      13$
3628                  000037    MICPC=MICPC+1
3629  015622  101452            .WORD   .$.
3630  015624                    MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
3631                  000040    MICPC=MICPC+1
3632  015624  061224            .WORD   .$.
3633  015626                    MOVE    INP0 <0>,OUT1 <CSR5>;BAD DATA...
3634                  000041    MICPC=MICPC+1
3635  015626  021005            .WORD   .$.
3636  015630                    MOVE    # 1,MEM                 ;TYPE OF ERROR...
3637                  000042    MICPC=MICPC+1
3638  015630  002401            .WORD   .$.
3639  015632                    MOVE    MEM,OUT1 <CSR3>         ;
3640                  000043    MICPC=MICPC+1
3641  015632  041223            .WORD   .$.
3642  015634                    MOVE    # 0,MEM         ;
3643                  000044    MICPC=MICPC+1
```

```
3644  015634  002400                .WORD   .$.
3645  015636                        MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
3646          000045                MICPC=MICPC+1
3647  015636  041227                .WORD   .$.
3648  015640                        CALL    ERÒR                    ;REPORT DATA ERROR.
3649  015640                        MOVE    # <MICPC+3),BREG
3650          000046                MICPC=MICPC+1
3651  015640  000450                .WORD   .$.
3652  015642                        SBR     ERÒR
3653          000047                MICPC=MICPC+1
3654  015642  104400                .WORD   .$.
3655  015644                        MOVE    SPAD <4>,BREG           ;RESTORE BREG...
3656          000050                MICPC=MICPC+1
3657  015644  060604                .WORD   .$.
3658  015646                        SBR     12$                     ;LOOP ON ERROR...
3659          000051                MICPC=MICPC+1
3660  015646  100433                .WORD   .$.
3661  015650                13$:    CALL    SCP1                    ;IS LOOP DATA SET.
3662  015650                        MOVE    # <MICPC+3),BREG
3663          000052                MICPC=MICPC+1
3664  015650  000454                .WORD   .$.
3665  015652                        SBR     SCP1
3666          000053                MICPC=MICPC+1
3667  015652  104427                .WORD   .$.
3668  015654                        MOVE    SPAD <4),BREG
3669          000054                MICPC=MICPC+1
3670  015654  060604                .WORD   .$.
3671  015656                        SBR     12$                     ;YES, DO IT.
3672          000055                MICPC=MICPC+1
3673  015656  100433                .WORD   .$.
3674  015660                        MOVE    SPAD <4),BREG
3675          000056                MICPC=MICPC+1
3676  015660  060604                .WORD   .$.
3677  015662                        SHFRT                           ;NO, CONTINUE...
3678          000057                MICPC=MICPC+1
3679  015662  061620                .WORD   .SBR!.SELB!.DBRSH
3680  015664                        BR7     12$                     ;
3681          000060                MICPC=MICPC+1
3682  015664  103433                .WORD   .$.
3683  015666                14$:
3684  015666                        CALL    SCPE
3685  015666                        MOVE    # <MICPC+3),BREG
3686          000061                MICPC=MICPC+1
3687  015666  000463                .WORD   .$.
3688  015670                        SBR     SCPE
3689          000062                MICPC=MICPC+1
3690  015670  104454                .WORD   .$.
3691  015672                        SBR     21$
3692          000063                MICPC=MICPC+1
3693  015672  100400                .WORD   .$.
3694  015674                SFLT1   OUTO,INPO,1,0
3695  015674                SXZ
3696
3697
3698                                ;***************************** TEST 10 ****************************
3699                                ;* MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
```

```
                                        ;* FLOAT A 1 THROUGH REGISTER OUTO (1)
                                        ;* FLOAT A 0THROUGH REGISTER OUTO (1)
015674                          $XZ
                                        ;;*****************************************************************
015674                          $TSTN
                                        ;  TEST 10
                                        ;-------------
015674  012737  000010  001202  TST10:  MOV     #10,$TSTNM              ; LOAD THE NO. OF THIS TEST
015702  012737  016100  001442          MOV     #TST11,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
015710  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
015714  015730                          MCT10
015716  104022                          ERROR   22                      ;TIME OUT ERROR...
015720  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
015724  000177  163512                  JMP     @NEXT                   ;GO TO NEXT TEST...
015730                          MCT10:
015730                          21$:
015730                                  MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
000000                                  MICPC=MICPC+1
015730  000400                          .WORD   .$.
015732                                  MOVE    BREG,SPAD (16)  ;FOR RETURN ADDRESS PURPOSES...
000001                                  MICPC=MICPC+1
015732  063236                          .WORD   .$.
015734                          $FLOT   OUTO,INPO,1,0,1,1$,2$,3$,4$
015734                          1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
000002                                  MICPC=MICPC+1
015734  000600                          .WORD   .$.
015736                          2$:
015736                                  MOVE    BREG,OUTO (1)   ;SET THE BIT.
000003                                  MICPC=MICPC+1
015736  062221                          .WORD   .$.
015740                                  MOVE    INPO (1),SPAD (0)       ;GET THE "FOUND" IN SCRATCH PAD.
000004                                  MICPC=MICPC+1
015740  023020                          .WORD   .$.
015742                                  MOVE    BREG,SPAD (4)
000005                                  MICPC=MICPC+1
015742  063224                          .WORD   .$.
015744                                  $IFEQ   BREG,SPAD (0)  3$       ;CHECK THE DATA...


015744                                  SUB2C   SPAD (0),BREG,NOP
000006                                  MICPC=MICPC+1
015744  060360                          .WORD   .$.
015746                                  BZ      3$
000007                                  MICPC=MICPC+1
015746  101422                          .WORD   .$.
015750                                  MOVE    BREG,OUT1 (CSR4)        ;GOOD DATA...
000010                                  MICPC=MICPC+1
015750  061224                          .WORD   .$.
015752                                  MOVE    INPO (1),OUT1 (CSR5);BAD DATA...
000011                                  MICPC=MICPC+1
015752  021025                          .WORD   .$.
015754                                  MOVE    # 1,MEM                 ;TYPE OF ERROR...
000012                                  MICPC=MICPC+1
015754  002401                          .WORD   .$.
```

```
015756                  MOVE    MEM,OUT1 <CSR3>         ;
                        MICPC=MICPC+1
000013                  .WORD   .$.
015756 041223           MOVE    # 1,MEM         ;
015760                  MICPC=MICPC+1
                        .WORD   .$.
000014                  MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
015760 002401           MICPC=MICPC+1
015762                  .WORD   .$.
                        CALL    ERROR                   ;REPORT DATA ERROR.
000015                  MOVE    # <MICPC+3>,BREG
015762 041227           MICPC=MICPC+1
015764                  .WORD   .$.
015764                  $BR     ERROR
                        MICPC=MICPC+1
000016                  .WORD   .$.
015764 000420           MOVE    $PAD <4>,BREG           ;RESTORE BREG...
015766                  MICPC=MICPC+1
                        .WORD   .$.
000017                  $BR     2$                      ;LOOP ON ERROR...
015766 104400           MICPC=MICPC+1
015770                  .WORD   .$.
                3$:     CALL    $CP1                    ;IS LOOP DATA SET.
000020                  MOVE    # <MICPC+3>,BREG
015770 060604           MICPC=MICPC+1
015772                  .WORD   .$.
                        $BR     $CP1
000021                  MICPC=MICPC+1
015772 100403           .WORD   .$.
015774                  MOVE    $PAD <4>,BREG
015774                  MICPC=MICPC+1
000022                  .WORD   .$.
015774 000424           $BR     2$                      ;YES, DO IT.
015776                  MICPC=MICPC+1
                        .WORD   .$.
000023                  MOVE    $PAD <4>,BREG
015776 104427           MICPC=MICPC+1
016000                  .WORD   .$.
                        SHFBRT                          ;NO, CONTINUE...
000024                  MICPC=MICPC+1
016000 060604           .WORD   .$BR!.SELB!.DBRSH
016002                  BB7     4$                      ;IS IT DONE?...
                        MICPC=MICPC+1
000025                  .WORD   .$.
016002 100403           $BR     2$                      ;NO, CONTINUE...
016004                  MICPC=MICPC+1
                        .WORD   .$.
000026                  4$:
016004 060604           $FLOT   OUT0,INP0,1,0,0,11$,12$,13$,14$
016006                  11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
                        MICPC=MICPC+1
000027                  .WORD   .$.
016006 061620           12$:
016010                  MOVE    BREG,OUT0 <1>   ;SET THE BIT.
                        MICPC=MICPC+1
000030                  .WORD   .$.
016010 103432
016012
000031
016012 100403
016014
016014
016014
000032
016014 000577
016016
016016
000033
016016 062221
```

```
3812  016020                    MOVE    INPO (1),SPAD (0)       ;GET THE "FOUND" IN SCRATCH PAD.
3813              000034        MICPC=MICPC+1
3814  016020      023020        .WORD   .$.
3815  016022                    MOVE    BREG,SPAD (4)
3816              000035        MICPC=MICPC+1
3817  016022      063224        .WORD   .$.
3818  016024                    $IFEQ   BREG,SPAD (0)  13$      ;CHECK THE DATA...
3819
3820
3821  016024                    SUB2C   SPAD (0),BREG,NOP
3822              000036        MICPC=MICPC+1
3823  016024      060360        .WORD   .$.
3824  016026                    BZ      13$
3825              000037        MICPC=MICPC+1
3826  016026      101452        .WORD   .$.
3827  016030                    MOVE    BREG,OUT1 (CSR4)        ;GOOD DATA...
3828              000040        MICPC=MICPC+1
3829  016030      061224        .WORD   .$.
3830  016032                    MOVE    INPO (1),OUT1 (CSR5);BAD DATA...
3831              000041        MICPC=MICPC+1
3832  016032      021025        .WORD   .$.
3833  016034                    MOVE    # 1,MEM                 ;TYPE OF ERROR...
3834              000042        MICPC=MICPC+1
3835  016034      002401        .WORD   .$.
3836  016036                    MOVE    MEM,OUT1 (CSR3)         ;
3837              000043        MICPC=MICPC+1
3838  016036      041223        .WORD   .$.
3839  016040                    MOVE    # 1,MEM         ;
3840              000044        MICPC=MICPC+1
3841  016040      002401        .WORD   .$.
3842  016042                    MOVE    MEM,OUT1 (CSR7)         ;REG. ADDRESS.
3843              000045        MICPC=MICPC+1
3844  016042      041227        .WORD   .$.
3845  016044                    CALL    ERROR                   ;REPORT DATA ERROR.
3846  016044                    MOVE    # (MICPC+3),BREG
3847              000046        MICPC=MICPC+1
3848  016044      000450        .WORD   .$.
3849  016046                    SBR     ERROR
3850              000047        MICPC=MICPC+1
      016046      104400        .WORD   .$.
      016050                    MOVE    SPAD (4),BREG           ;RESTORE BREG...
              000050            MICPC=MICPC+1
3854  016050      060604        .WORD   .$.
3855  016052                    SBR     12$                     ;LOOP ON ERROR...
3856              000051        MICPC=MICPC+1
3857  016052      100433        .WORD   .$.
3858  016054
3859  016054                    CALL    SCP1                    ;IS LOOP DATA SET.
              000052            MOVE    # (MICPC+3),BREG
      016054      000454        MICPC=MICPC+1
      016056                    .WORD   .$.
              000053            SBR     SCP1
      016056      104427        MICPC=MICPC+1
      016060                    .WORD   .$.
              000054            MOVE    SPAD (4),BREG
3867  016060      060604        MICPC=MICPC+1
                                .WORD   .$.
```

```
016062                          SBR     12$                     ;YES, DO IT.
                        000055  MICPC=MICPC+1
016062  100433                  .WORD   .$.
016064                          MOVE    SPAD (4),BREG
                        000056  MICPC=MICPC+1
016064  060604                  .WORD   .$.
016066                          SHFBRT                          ;NO, CONTINUE...
                        000057  MICPC=MICPC+1
016066  061620                  .WORD   .SBR!.SELB!.DBRSH
016070                          BR7     12$                     ;
                        000060  MICPC=MICPC+1
016070  103433                  .WORD   .$.
016072                  14$:
016072                          CALL    SCPE
016072                          MOVE    # (MICPC+3),BREG
                        000061  MICPC=MICPC+1
016072  000463                  .WORD   .$.
016074                          SBR     SCPE
                        000062  MICPC=MICPC+1
016074  104454                  .WORD   .$.
016076                          SBR     21$
                        000063  MICPC=MICPC+1
016076  100400                  .WORD   .$.
016100                  $FLT1   OUTO,INPO,2,0
016100                  $X2


                        ;************************** TEST 11 ****************************
                        ;* MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
                        ;* FLOAT A 1 THROUGH REGISTER OUTO (2)
                        ;* FLOAT A 0THROUGH REGISTER OUTO (2)
016100                  $XZ
                        ;;*************************************************************


016100                  $TSTN
                        ;  TEST 11
                        ;--------------
016100  012737  000011  001202  TST11:  MOV     #11,$TSTNM              ; LOAD THE NO. OF THIS TEST
016106  012737  016304  001442          MOV     #TST12,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
016114  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
016120  016134                          MCT11
016122  104022                          ERROR   22                      ;TIME OUT ERROR...
016124  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
016130  000177  163306                  JMP     @NEXT                   ;GO TO NEXT TEST...
016134                  MCT11:
016134                  21$:
016134                          MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
                        000000  MICPC=MICPC+1
016134  000400                  .WORD   .$.
016136                          MOVE    BREG,SPAD (16)  ;FOR RETURN ADDRESS PURPOSES...
                        000001  MICPC=MICPC+1
016136  063236                  .WORD   .$.
016140                  $FLOT   OUTO,INPO,2,0,1,1$,2$,3$,4$
016140                  1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
                        000002  MICPC=MICPC+1
```

```
016140 000600                    .WORD   .$.
016142                     2$:
016142                           MOVE    BREG,OUT0 <2>   ;SET THE BIT.
000003                           MICPC=MICPC+1
016142 062222                    .WORD   .$.
016144                           MOVE    INP0 <2>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
000004                           MICPC=MICPC+1
016144 023040                    .WORD   .$.
016146                           MOVE    BREG,SPAD <4>
000005                           MICPC=MICPC+1
016146 063224                    .WORD   .$.
016150                           $IFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...

016150                           SUB2C   SPAD <0>,BREG,NOP
000006                           MICPC=MICPC+1
016150 060360                    .WORD   .$.
016152                           BZ      3$
000007                           MICPC=MICPC+1
016152 101422                    .WORD   .$.
016154                           MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
000010                           MICPC=MICPC+1
016154 061224                    .WORD   .$.
016156                           MOVE    INP0 <2>,OUT1 <CSR5>;BAD DATA...
000011                           MICPC=MICPC+1
016156 021045                    .WORD   .$.
016160                           MOVE    # 1,MEM                 ;TYPE OF ERROR...
000012                           MICPC=MICPC+1
016160 002401                    .WORD   .$.
016162                           MOVE    MEM,OUT1 <CSR3>         ;
000013                           MICPC=MICPC+1
016162 041223                    .WORD   .$.
016164                           MOVE    # 2,MEM         ;
000014                           MICPC=MICPC+1
016164 002402                    .WORD   .$.
016166                           MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
000015                           MICPC=MICPC+1
016166 041227                    .WORD   .$.
016170                           CALL    ERROR                   ;REPORT DATA ERROR.
016170                           MOVE    # <MICPC+3>,BREG
000016                           MICPC=MICPC+1
016170 000420                    .WORD   .$.
016172                           $BR     ERROR
000017                           MICPC=MICPC+1
016172 104400                    .WORD   .$.
016174                           MOVE    SPAD <4>,BREG           ;RESTORE BREG...
000020                           MICPC=MICPC+1
016174 060604                    .WORD   .$.
016176                           $BR     2$                      ;LOOP ON ERROR...
000021                           MICPC=MICPC+1
016176 100403                    .WORD   .$.
016200                     3$:   CALL    SCP1                    ;IS LOOP DATA SET.
016200                           MOVE    # <MICPC+3>,BREG
000022                           MICPC=MICPC+1
016200 000424                    .WORD   .$.
016202                           $BR     SCP1
```

```
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    SPAD (4),BREG
                        MICPC=MICPC+1
                        .WORD   .$.
                        SBR     2$                      ;YES, DO IT.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    SPAD (4),BREG
                        MICPC=MICPC+1
                        .WORD   .$.
                        SHFBRT                          ;NO, CONTINUE...
                        MICPC=MICPC+1
                        .WORD   SBR!.SELB!.DBRSH
                        BBZ     4$                      ;IS IT DONE?...
                        MICPC=MICPC+1
                        .WORD   .$.
                        SBR     2$                      ;NO, CONTINUE...
                        MICPC=MICPC+1
                        .WORD   .$.
                4$:
                SFLOT   OUT0,INP0,2,0,0,11$,12$,13$,14$
                11$:    MOVE    # 177,BREG      ;START WITH BIT 7.
                        MICPC=MICPC+1
                        .WORD   .$.
                12$:
                        MOVE    BREG,OUT0 (2)   ;SET THE BIT.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    INP0 (2),SPAD (0)       ;GET THE "FOUND" IN SCRATCH PAD.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    BREG,SPAD (4)
                        MICPC=MICPC+1
                        .WORD   .$.
                        $IFEQ   BREG,SPAD (0)  13$      ;CHECK THE DATA...


                        SUB2C   SPAD (0),BREG,NOP
                        MICPC=MICPC+1
                        .WORD   .$.
                        BZ      13$
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    BREG,OUT1 (CSR4)        ;GOOD DATA...
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    INP0 (2),OUT1 (CSR5);BAD DATA...
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    # 1,MEM                 ;TYPE OF ERROR...
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    MEM,OUT1 (CSR3)         ;
                        MICPC=MICPC+1
                        .WORD   .$.
```

```
4036  016244
4037                 000044
4038  016244         002402
4039  016246
4040                 000045
4041  016246         041227
4042  016250
4043  016250
4044                 000046
4045  016250         000450
4046  016252
4047                 000047
4048  016252         104400
4049  016254
4050                 000050
4051  016254         060604
4052  016256
4053                 000051
4054  016256         100433
4055  016260
4056  016260
4057                 000052
4058  016260         000454
4059  016262
4060                 000053
4061  016262         104427
4062  016264
4063                 000054
4064  016264         060604
4065  016266
4066                 000055
4067  016266         100433
4068  016270
4069                 000056
4070  016270         060604
4071  016272
4072                 000057
4073  016272         061620
4074  016274
4075                 000060
4076  016274         103433
4077  016276
4078  016276
4079  016276
4080                 000061
4081  016276         000463
4082  016300
4083                 000062
4084  016300         104454
4085  016302
4086                 000063
4087  016302         100400
4088  016304
4089  016304
4090
4091
```

```
                MOVE    #.2,MEM         ;
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,OUT1 (CSR7)         ;REG. ADDRESS.
                MICPC=MICPC+1
                .WORD   .$.
                CALL    ERROR           ;REPORT DATA ERROR.
                MOVE    # (MICPC+3),BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     ERROR
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD (4),BREG           ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     13$             ;LOOP ON ERROR...
                MICPC=MICPC+1
                .WORD   .$.
13$:            CALL    SCP1            ;IS LOOP DATA SET.
                MOVE    # (MICPC+3),BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     SCP1
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD (4),BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     12$                     ;YES, DO IT.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD (4),BREG
                MICPC=MICPC+1
                .WORD   .$.
                SHFBRT                          ;NO, CONTINUE...
                MICPC=MICPC+1
                .WORD   .SBR!.SELB!.DBRSH
                BR7     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
                .WORD   .$.
$FLT1           OUT0,INP0,3,0
$XZ
```

```
                                        ;*************************** TEST 12 ***************************
                                        ;* MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
                                        ;* FLOAT A 1 THROUGH REGISTER OUTO <3>
                                        ;* FLOAT A 0THROUGH REGISTER OUTO <3>
016304                          SXZ
                                        ;;*************************************************************************

016304                          STSTN
                                        ;  TEST 12
                                        ;-------------
016304 012737 000012 001202     TST12:  MOV     #12,$TSTNM              ; LOAD THE NO. OF THIS TEST
016312 012737 016510 001442             MOV     #TST13,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
016320 004737 035536                    JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
016324 016340                           MCT12
016326 104022                           ERROR   22                      ;TIME OUT ERROR...
016330 012706 001200                    MOV     #STACK,SP               ;RESET STACK...
016334 000177 163102                    JMP     @NEXT                   ;GO TO NEXT TEST...
016340                          MCT12:
016340                          21$:
016340                                  MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
000000                                  MICPC=MICPC+1
016340 000400                           .WORD   .$.
016342                                  MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
000001                                  MICPC=MICPC+1
016342 063236                           .WORD   .$.
016344                          SFLOT   OUTD,INPD,3,0,1,1$,2$,3$,4$
016344                          1$:     MOVE    # 200,BREG      ;START WITH BIT 7.
000002                                  MICPC=MICPC+1
016344 000600                           .WORD   .$.
016346                          2$:
016346                                  MOVE    BREG,OUTD <3>   ;SET THE BIT.
000003                                  MICPC=MICPC+1
016346 062223                           .WORD   .$.
016350                                  MOVE    INPD <3>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
000004                                  MICPC=MICPC+1
016350 023060                           .WORD   .$.
016352                                  MOVE    BREG,SPAD <4>
000005                                  MICPC=MICPC+1
016352 063224                           .WORD   .$.
016354                                  $IFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...


016354                                  SUB2C   SPAD <0>,BREG,NOP
000006                                  MICPC=MICPC+1
016354 060360                           .WORD   .$.
016356                                  BZ      3$
000007                                  MICPC=MICPC+1
016356 101422                           .WORD   .$.
016360                                  MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
000010                                  MICPC=MICPC+1
016360 061224                           .WORD   .$.
016362                                  MOVE    INPD <3>,OUT1 <CSR5>;BAD DATA...
000011                                  MICPC=MICPC+1
016362 021065                           .WORD   .$.
016364                                  MOVE    # 1,MEM                 ;TYPE OF ERROR...
```

```
4148                    000012
4149  016364  002401
4150  016366
4151                    000013
4152  016366  041223
4153  016370
4154                    000014
4155  016370  002403
4156  016372
4157                    000015
4158  016372  041227
4159  016374
4160  016374
4161                    000016
4162  016374  000420
4163  016376
4164                    000017
4165  016376  104400
4166  016400
4167                    000020
4168  016400  060604
4169  016402
4170                    000021
4171  016402  100403
4172  016404
4173  016404
4174                    000022
4175  016404  000424
4176  016406
4177                    000023
4178  016406  104427
4179  016410
4180                    000024
4181  016410  060604
4182  016412
4183                    000025
4184  016412  100403
4185  016414
4186                    000026
4187  016414  060604
4188  016416
4189                    000027
4190  016416  061620
4191  016420
4192                    000030
4193  016420  103432
4194  016422
4195                    000031
4196  016422  100403
4197  016424
4198  016424
4199  016424
4200                    000032
4201  016424  000577
4202  016426
4203  016426
```

```
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,OUT1 <CSR3>         ;
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 3,MEM                 ;
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
                MICPC=MICPC+1
                .WORD   .$.
                CALL    EROR                    ;REPORT DATA ERROR.
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     EROR
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,BREG           ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     2$                      ;LOOP ON ERROR...
                MICPC=MICPC+1
                .WORD   .$.
3$:             CALL    SCP1                    ;IS LOOP DATA SET.
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     SCP1
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     2$                      ;YES, DO IT.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SHFRT                           ;NO, CONTINUE...
                MICPC=MICPC+1
                .WORD   .SBR!.SELB!.DBRSH
                BR7     4$                      ;IS IT DONE?...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     2$                      ;NO, CONTINUE...
                MICPC=MICPC+1
                .WORD   .$.
4$:
SFLOT   OUT0,INP0,3,0,0,11$,12$,13$,14$
11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
                MICPC=MICPC+1
                .WORD   .$.
12$:
                MOVE    BREG,OUT0 <3>   ;SET THE BIT.
```

```
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    INPO (3),SPAD (0)       ;GET THE "FOUND" IN SCRATCH PAD.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    BREG,SPAD (4)
                                MICPC=MICPC+1
                                .WORD   .$.
                                $IFEQ   BREG,SPAD (0)  13$      ;CHECK THE DATA...


                                SUB2C   SPAD (0),BREG,NOP
                                MICPC=MICPC+1
                                .WORD   .$.
                                BR      12$
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    BREG,OUT1 (CSR4)        ;GOOD DATA...
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    INPO (3),OUT1 (CSR5);BAD DATA...
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # 1,MEM                 ;TYPE OF ERROR...
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    MEM,OUT1 (CSR3)         ;
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # 3,MEM         ;
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    MEM,OUT1 (CSR7)         ;REG. ADDRESS.
                                MICPC=MICPC+1
                                .WORD   .$.
                                CALL    ERROR                   ;REPORT DATA ERROR.
                                MOVE    # (MICPC+3),BREG
                                MICPC=MICPC+1
                                .WORD   .$.
                                SBR     ERROR
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    SPAD (4),BREG           ;RESTORE BREG...
                                MICPC=MICPC+1
                                .WORD   .$.
                                SBR     12$                     ;LOOP ON ERROR...
                                MICPC=MICPC+1
                                .WORD   .$.
                        13$:    CALL    SCP1                    ;IS LOOP DATA SET.
                                MOVE    # (MICPC+3),BREG
                                MICPC=MICPC+1
                                .WORD   .$.
                                SBR     SCP1
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    SPAD (4),BREG
```

```
                        000054                  MICPC=MICPC+1
016470  060604          .WORD   .$.
016472                  SBR     i2$             ;YES, DO IT.
                        000055                  MICPC=MICPC+1
016472  100433          .WORD   .$.
016474                  MOVE    SPAD (4),BREG
                        000056                  MICPC=MICPC+1
016474  060604          .WORD   .$.
016476                  SHFRT                   ;NO, CONTINUE...
                        000057                  MICPC=MICPC+1
016476  061620          .WORD   .SBR!.SELB!.DBRSH
016600                  BR7     i2$             ;
                        000060                  MICPC=MICPC+1
016600  103433          .WORD   .$.
016602                  14$:
016602                  CALL    SCPE
016602                  MOVE    @ (MICPC+3),BREG
                        000061                  MICPC=MICPC+1
016602  000463          .WORD   .$.
016604                  SBR     SCPE
                        000062                  MICPC=MICPC+1
016604  104454          .WORD   .$.
016606                  SBR     21$
                        000063                  MICPC=MICPC+1
016606  100400          .WORD   .$.
016610                  5FLT1   OUTO,INPO,4,0
016510                  SXZ


                                ;############################ TEST 13 ############################
                                ;# MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
                                ;# FLOAT A 1 THROUGH REGISTER OUTO (4)
                                ;# FLOAT A 0THROUGH REGISTER OUTO (4)
016510                  SXZ
                                ;:#################################################################
016510                  STSTN
                                ;  TEST 13
                                ;-------------
016510  012737  000013  001202  TST13:  MOV     #13,STSTNM              ; LOAD THE NO. OF THIS TEST
016516  012737  016714  001442          MOV     #TST14,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
016524  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
016530  016544                          MCT13
016532  104022                          ERROR   22                      ;TIME OUT ERROR...
016534  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
016540  000177  162676                  JMP     @NEXT                   ;GO TO NEXT TEST...
016544                  MCT13:
016544                  21$:
016544                  MOVE    @ 0,BREG        ;SET TO CLEAR SPAD 16
                        000000                  MICPC=MICPC+1
016544  000400          .WORD   .$.
016546                  MOVE    BREG,SPAD (16)  ;FOR RETURN ADDRESS PURPOSES...
                        000001                  MICPC=MICPC+1
016546  063236          .WORD   .$.
016550                  SFLOT   OUTO,INPO,4,0,1,1$,2$,3$,4$
```

```
4316  016550                  1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
4317          000002                 MICPC=MICPC+1
4318  016550  000600                 .WORD   .$.
4319  016552                  2$:
4320  016552
4321          000003                 MOVE    BREG,OUTO <4>   ;SET THE BIT.
4322  016552  062224                 MICPC=MICPC+1
4323  016554                         .WORD   .$.
4324          000004                 MOVE    INPO <4>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
4325  016554  023100                 MICPC=MICPC+1
4326  016556                         .WORD   .$.
4327          000005                 MOVE    BREG,SPAD <4>
4328  016556  063224                 MICPC=MICPC+1
4329  016560                         .WORD   .$.
4330                                 SIFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...
4331
4332  016560                         SUB2C   SPAD <0>,BREG,NOP
4333          000006                 MICPC=MICPC+1
4334  016560  060360                 .WORD   .$.
4335  016562                         BZ      3$
4336          000007                 MICPC=MICPC+1
4337  016562  101422                 .WORD   .$.
4338  016564                         MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
4339          000010                 MICPC=MICPC+1
4340  016564  061224                 .WORD   .$.
4341  016566                         MOVE    INPO <4>,OUT1 <CSR5>;BAD DATA...
4342          000011                 MICPC=MICPC+1
4343  016566  021105                 .WORD   .$.
4344  016570                         MOVE    # 1,MEM                 ;TYPE OF ERROR...
4345          000012                 MICPC=MICPC+1
4346  016570  002401                 .WORD   .$.
4347  016572                         MOVE    MEM,OUT1 <CSR3>         ;
4348          000013                 MICPC=MICPC+1
4349  016572  041223                 .WORD   .$.
4350  016574                         MOVE    # 4,MEM         ;
4351          000014                 MICPC=MICPC+1
4352  016574  002404                 .WORD   .$.
4353  016576                         MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
4354          000015                 MICPC=MICPC+1
4355  016576  041227                 .WORD   .$.
4356  016600                         CALL    ERROR                   ;REPORT DATA ERROR.
4357  016600                         MOVE    # <MICPC+3>,BREG
4358          000016                 MICPC=MICPC+1
4359  016600  000420                 .WORD   .$.
4360  016602                         SBR     ERROR
4361          000017                 MICPC=MICPC+1
4362  016602  104400                 .WORD   .$.
4363  016604                         MOVE    SPAD <4>,BREG           ;RESTORE BREG...
4364          000020                 MICPC=MICPC+1
4365  016604  060604                 .WORD   .$.
4366  016606                         SBR     2$                      ;LOOP ON ERROR...
4367          000021                 MICPC=MICPC+1
4368  016606  100403                 .WORD   .$.
4369  016610                  3$:    CALL    SCP1                    ;IS LOOP DATA SET.
4370  016610                         MOVE    # <MICPC+3>,BREG
4371          000022                 MICPC=MICPC+1
```

```
4372  016610  000424          .WORD   .$.
4373  016612                  SBR     SCP1
4374          000023          MICPC=MICPC+1
4375  016612  104427          .WORD   .$.
4376  016614                  MOVE    SPAD (4),BREG
4377          000024          MICPC=MICPC+1
4378  016614  060604          .WORD   .$.
4379  016616                  SBR     2S                      ;YES, DO IT.
4380          000025          MICPC=MICPC+1
4381  016616  100403          .WORD   .$.
4382  016620                  MOVE    SPAD (4),BREG
4383          000026          MICPC=MICPC+1
4384  016620  060604          .WORD   .$.
4385  016622                  SHFBRT                          ;NO, CONTINUE...
4386          000027          MICPC=MICPC+1
4387  016622  061620          .WORD   .SBR!.SELB!.DBRSH
4388  016624                  BR7     4S                      ;IS IT DONE?...
4389          000030          MICPC=MICPC+1
4390  016624  103432          .WORD   .$.
4391  016626                  SBR     2S                      ;NO, CONTINUE...
4392          000031          MICPC=MICPC+1
4393  016626  100403          .WORD   .$.
4394  016630                  4S:
4395  016630                  SFLOT   OUT0,INP0,4,0,0,11$,12$,13$,14$
4396  016630                  11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
4397          000032          MICPC=MICPC+1
4398  016630  000577          .WORD   .$.
4399  016632                  12$:
4400  016632                  MOVE    BREG,OUT0 (4)   ;SET THE BIT.
4401          000033          MICPC=MICPC+1
4402  016632  062224          .WORD   .$.
4403  016634                  MOVE    INP0 (4),SPAD (0)       ;GET THE "FOUND" IN SCRATCH PAD.
4404          000034          MICPC=MICPC+1
4405  016634  023100          .WORD   .$.
4406  016636                  MOVE    BREG,SPAD (4)
4407          000035          MICPC=MICPC+1
4408  016636  063224          .WORD   .$.
4409  016640                  SIFEQ   BREG,SPAD (0)  13$      ;CHECK THE DATA...
4410
4411
4412  016640                  SUB2C   SPAD (0),BREG,NOP
4413          000036          MICPC=MICPC+1
4414  016640  060360          .WORD   .$.
4415  016642                  BZ      13$
4416          000037          MICPC=MICPC+1
4417  016642  101452          .WORD   .$.
4418  016644                  MOVE    BREG,OUT1 (CSR4)        ;GOOD DATA...
4419          000040          MICPC=MICPC+1
4420  016644  061224          .WORD   .$.
4421  016646                  MOVE    INP0 (4),OUT1 (CSR5);BAD DATA...
4422          000041          MICPC=MICPC+1
4423  016646  021105          .WORD   .$.
4424  016650                  MOVE    # 1,MEM                 ;TYPE OF ERROR...
4425          000042          MICPC=MICPC+1
4426  016650  002401          .WORD   .$.
4427  016652                  MOVE    MEM,OUT1 (CSR3)         ;
```

```
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    # 4,MEM          ;
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
                        MICPC=MICPC+1
                        .WORD   .$.
                        CALL    ERROR                   ;REPORT DATA ERROR.
                        MOVE    # <MICPC+3>,BREG
                        MICPC=MICPC+1
                        .WORD   .$.
                        SBR     ERROR
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    SPAD <4>,BREG           ;RESTORE BREG...
                        MICPC=MICPC+1
                        .WORD   .$.
                        SBR     12$                     ;LOOP ON ERROR...
                        MICPC=MICPC+1
                        .WORD   .$.
135:                    CALL    SCP1                    ;IS LOOP DATA SET.
                        MOVE    # <MICPC+3>,BREG
                        MICPC=MICPC+1
                        .WORD   .$.
                        SBR     SCP1
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    SPAD <4>,BREG
                        MICPC=MICPC+1
                        .WORD   .$.
                        SBR     13$                     ;YES, DO IT.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    SPAD <4>,BREG
                        MICPC=MICPC+1
                        .WORD   .$.
                        SHFBRT                          ;NO, CONTINUE...
                        MICPC=MICPC+1
                        .WORD   .SBR!.SELB!.DBRSH
                        BR7     14$             ;
                        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
                        .WORD   .$.
$FLT1   OUT0,INP0,5,0
$XZ
```

```
4484
4485
4486                                    ;*************************** TEST 14 ****************************
4487                                    ;* MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
4488                                    ;* FLOAT A 1 THROUGH REGISTER OUTO (S)
4489                                    ;* FLOAT A 0THROUGH REGISTER OUTO (S)
4490  016714                    $XZ
4491                                    ;:************************************************************************
4492
4493  016714                    $TSTN
4494                                    ;  TEST 14
4495                                    ;--------------
4496  016714  012737  000014  001202  TST14:  MOV     #14,$TSTNM              ; LOAD THE NO. OF THIS TEST
4497  016722  012737  017120  001442          MOV     #TST1S,NEXT             ; POINT TO THE START OF NEXT TEST.
4498                                                                  ;R1 CONTAINS BASE KMC11 ADDRESS
4499  016730  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
4500  016734  016750                          MCT14
4501  016736  104022                          ERROR   22                      ;TIME OUT ERROR...
4502  016740  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
4503  016744  000177  162472                  JMP     @NEXT                   ;GO TO NEXT TEST...
4504  016750                    MCT14:
4505  016750                    21$:
4506  016750                            MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
4507            000000                  MICPC=MICPC+1
4508  016750   000400                   .WORD   .$.
4509  016752                            MOVE    BREG,SPAD (16)  ;FOR RETURN ADDRESS PURPOSES...
4510            000001                  MICPC=MICPC+1
4511  016752   063236                   .WORD   .$.
4512  016754                    $FLOT   OUTO,INPO,5,0,1,1$,2$,3$,4$
4513  016754                    1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
4514            000002                  MICPC=MICPC+1
4515  016754   000600                   .WORD   .$.
4516  016756                    2$:
4517  016756                            MOVE    BREG,OUTO (5)   ;SET THE BIT.
4518            000003                  MICPC=MICPC+1
4519  016756   062225                   .WORD   .$.
4520  016760                            MOVE    INPO (5),SPAD (0)       ;GET THE "FOUND" IN SCRATCH PAD.
4521            000004                  MICPC=MICPC+1
4522  016760   023120                   .WORD   .$.
4523  016762                            MOVE    BREG,SPAD (4)
4524            000005                  MICPC=MICPC+1
4525  016762   063224                   .WORD   .$.
4526  016764                            SIFEQ   BREG,SPAD (0)  3$       ;CHECK THE DATA...
4527
4528  016764
4529                                    SUB2C   SPAD (0),BREG,NOP
4530            000006                  MICPC=MICPC+1
4531  016764   060360                   .WORD   .$.
4532  016766                            BZ      3$
4533            000007                  MICPC=MICPC+1
4534  016766   101422                   .WORD   .$.
4535  016770                            MOVE    BREG,OUT1 (CSR4)        ;GOOD DATA...
4536            000010                  MICPC=MICPC+1
4537  016770   061224                   .WORD   .$.
4538  016772                            MOVE    INPO (5),OUT1 (CSR5);BAD DATA...
4539            000011                  MICPC=MICPC+1
```

```
4540  016772  021125                .WORD   .$.
4541  016774                        MOVE    # 1,MEM                 ;TYPE OF ERROR...
4542          000012                MICPC=MICPC+1
4543  016774  002401                .WORD   .$.
4544  016776                        MOVE    MEM,OUT1 <CSR3>         ;
4545          000013                MICPC=MICPC+1
4546  016776  041223                .WORD   .$.
4547  017000                        MOVE    # 5,MEM         ;
4548          000014                MICPC=MICPC+1
4549  017000  002405                .WORD   .$.
4550  017002                        MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
4551          000015                MICPC=MICPC+1
4552  017002  041227                .WORD   .$.
4553  017004                        CALL    EROR                    ;REPORT DATA ERROR.
4554  017004                        MOVE    # <MICPC+3),BREG
4555          000016                MICPC=MICPC+1
4556  017004  000420                .WORD   .$.
4557  017006                        SBR     EROR
4558          000017                MICPC=MICPC+1
4559  017006  104400                .WORD   .$.
4560  017010                        MOVE    SPAD <4),BREG           ;RESTORE BREG...
4561          000020                MICPC=MICPC+1
4562  017010  060604                .WORD   .$.
4563  017012                        SBR     2$                      ;LOOP ON ERROR...
4564          000021                MICPC=MICPC+1
4565  017012  100403                .WORD   .$.
4566  017014                3$:     CALL    SCP1                    ;IS LOOP DATA SET.
4567  017014                        MOVE    # <MICPC+3),BREG
4568          000022                MICPC=MICPC+1
4569  017014  000424                .WORD   .$.
4570  017016                        SBR     SCP1
4571          000023                MICPC=MICPC+1
4572  017016  104427                .WORD   .$.
4573  017020                        MOVE    SPAD <4),BREG
4574          000024                MICPC=MICPC+1
4575  017020  060604                .WORD   .$.
4576  017022                        SBR     2$                      ;YES, DO IT.
4577          000025                MICPC=MICPC+1
4578  017022  100403                .WORD   .$.
4579  017024                        MOVE    SPAD <4),BREG
4580          000026                MICPC=MICPC+1
4581  017024  060604                .WORD   .$.
4582  017026                        SHFRT                           ;NO, CONTINUE...
4583          000027                MICPC=MICPC+1
4584  017026  061620                .WORD   .SBR!.SELB!.DBRSH
4585  017030                        BR?     4$                      ;IS IT DONE?...
4586          000030                MICPC=MICPC+1
4587  017030  103432                .WORD   .$.
4588  017032                        SBR     2$                      ;NO, CONTINUE...
4589          000031                MICPC=MICPC+1
4590  017032  100403                .WORD   .$.
4591  017034                4$:
4592  017034                SFLOT   OUTO,INPO,5,0,0,11$,12$,13$,14$
4593  017034                11$:    MOVE    # 177,BREG      ;START WITH BIT 7.
4594          000032                MICPC=MICPC+1
4595  017034  000577                .WORD   .$.
```

```
4596  017036
4597  017036
4598                000033
4599  017036  062225
4600  017040
4601                000034
4602  017040  023120
4603  017042
4604                000035
4605  017042  063224
4606  017044
4607
4608
4609  017044
4610                000036
4611  017044  060360
4612  017046
4613                000037
4614  017046  101452
4615  017050
4616                000040
4617  017050  061224
4618  017052
4619                000041
4620  017052  021125
4621  017054
4622                000042
4623  017054  002401
4624  017056
4625                000043
4626  017056  041223
4627  017060
4628                000044
4629  017060  002405
4630  017062
4631                000045
4632  017062  041227
4633  017064
4634  017064
4635                000046
4636  017064  000450
4637  017066
4638                000047
4639  017066  104400
4640  017070
4641                000050
4642  017070  060604
4643  017072
4644                000051
4645  017072  100433
4646  017074
4647  017074
4648                000052
4649  017074  000454
4650  017076
465:                000053
```

```
12$:
        MOVE    BREG,OUT0 (5)   ;SET THE BIT.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    INP0 (5),SPAD (0)       ;GET THE "FOUND" IN SCRATCH PAD.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,SPAD (4)
        MICPC=MICPC+1
        .WORD   .$.
        SIFEQ   BREG,SPAD (0)  13$      ;CHECK THE DATA...

        SUB2C   SPAD (0),BREG,NOP
        MICPC=MICPC+1
        .WORD   .$.
        BZ      13$
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 (CSR4)        ;GOOD DATA...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    INP0 (5),OUT1 (CSR5);BAD DATA...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 1,MEM                 ;TYPE OF ERROR...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,OUT1 (CSR3)         ;
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 5,MEM          ;
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,OUT1 (CSR7)         ;REG. ADDRESS.
        MICPC=MICPC+1
        .WORD   .$.
        CALL    ERROR                   ;REPORT DATA ERROR.
        MOVE    # (MICPC+3),BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     ERROR
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD (4),BREG           ;RESTORE BREG...
        MICPC=MICPC+1
        .WORD   .$.
        SBR     12$                     ;LOOP ON ERROR...
        MICPC=MICPC+1
        .WORD   .$.
13$:    CALL    SCP1                    ;IS LOOP DATA SET.
        MOVE    # (MICPC+3),BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     SCP1
        MICPC=MICPC+1
```

```
4652  017076  104427          .WORD   .$.
4653  017100                  MOVE    SPAD (4),BREG
4654                  000054
4655  017100  060604           MICPC=MICPC+1
4656  017102                  .WORD   .$.
4657                  000055    SBR     12$                     ;YES, DO IT.
4658  017102  100433           MICPC=MICPC+1
4659  017104                  .WORD   .$.
4660                  000056    MOVE    SPAD (4),BREG
4661  017104  060604           MICPC=MICPC+1
4662  017106                  .WORD   .$.
4663                  000057    SHFBRT                          ;NO, CONTINUE...
4664  017106  061620           MICPC=MICPC+1
4665  017110                  .WORD   .SBR!.SELB!.DBRSH
4666                  000060    BR      12$                     ;
4667  017110  103433           MICPC=MICPC+1
4668  017112                  .WORD   .$.
4669  017112                  14$:
4670  017112                  CALL    SCPE
4671                  000061    MOVE    # (MICPC+3),BREG
4672  017112  000463           MICPC=MICPC+1
4673  017114                  .WORD   .$.
4674                  000062    SBR     SCPE
4675  017114  104454           MICPC=MICPC+1
4676  017116                  .WORD   .$.
4677                  000063    SBR     21$
4678  017116  100400           MICPC=MICPC+1
4679  017120                  .WORD   .$.
4680  017120                  $FLT1   OUTO,INPO,6,0
4681                          $XZ
4682
4683                          ;*************************** TEST 15 ****************************
4684                          ;* MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
4685                          ;* FLOAT A 1 THROUGH REGISTER OUTO (6)
4686                          ;* FLOAT A 0THROUGH REGISTER OUTO (6)
4687  017120                  $XZ
4688                          ;;************************************************************************
4689
4690  017120                  $TSTN
4691                          ;  TEST 15
4692                          ;--------------
4693  017120  012737  000015  001202  TST15:  MOV     #15,$TSTNM              ; LOAD THE NO. OF THIS TEST
4694  017126  012737  017324  001442          MOV     #TST16,NEXT             ; POINT TO THE START OF NEXT TEST.
4695                                                                  ;R1 CONTAINS BASE KMC11 ADDRESS
4696  017134  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WRIT.
4697  017140  017154                          MCT15
4698  017142  104022                          ERROR   22                      ;TIME OUT ERROR...
4699  017144  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
4700  017150  000177  162266                  JMP     @NEXT                   ;GO TO NEXT TEST...
4701  017154                          MCT15:
4702  017154                          21$:
4703  017154                          MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
4704                  000000          MICPC=MICPC+1
4705  017154  000400                  .WORD   .$.
4706  017156                          MOVE    BREG,SPAD (16)  ;FOR RETURN ADDRESS PURPOSES...
4707                  000001          MICPC=MICPC+1
```

```
4708  017156  063236                  .WORD   .$
4709  017160                  SELOT   AUTO,INPO,0,1,1$,2$,3$,4$
4710  017160                  1$:  MOVE  # 200,BREG    ;START WITH BIT 7.
4711          000002                  MICPC=MICPC+1
4712  017160  000600                  .WORD   .$.
4713  017162                  2$:
4714  017162                          MOVE    BREG,OUT0 (6)   ;SET THE BIT.
4715          000003                  MICPC=MICPC+1
4716  017162  062226                  .WORD   .$.
4717  017164                          MOVE    INPO (6),SPAD (0)       ;GET THE "FOUND" IN SCRATCH PAD.
4718          000004                  MICPC=MICPC+1
4719  017164  023140                  .WORD   .$.
4720  017166                          MOVE    BREG,SPAD (4)
4721          000005                  MICPC=MICPC+1
4722  017166  063224                  .WORD   .$.
4723  017170                          SIFEQ   BREG,SPAD (0)  3$       ;CHECK THE DATA...
4724
4725
4726  017170                          SUB2C   SPAD (0),BREG,NOP
4727          000006                  MICPC=MICPC+1
4728  017170  060360                  .WORD   .$.
4729  017172                          BZ      3$
4730          000007                  MICPC=MICPC+1
4731  017172  101422                  .WORD   .$.
4732  017174                          MOVE    BREG,OUT1 (CSR4)        ;GOOD DATA...
4733          000010                  MICPC=MICPC+1
4734  017174  061224                  .WORD   .$.
4735  017176                          MOVE    INPO (6),OUT1 (CSR5);BAD DATA...
4736          000011                  MICPC=MICPC+1
4737  017176  021145                  .WORD   .$.
4738  017200                          MOVE    # 1,MEM                 ;TYPE OF ERROR...
4739          000012                  MICPC=MICPC+1
4740  017200  002401                  .WORD   .$.
4741  017202                          MOVE    MEM,OUT1 (CSR3)         ;
4742          000013                  MICPC=MICPC+1
4743  017202  041223                  .WORD   .$.
4744  017204                          MOVE    # 6,MEM         ;
4745          000014                  MICPC=MICPC+1
4746  017204  002406                  .WORD   .$.
4747  017206                          MOVE    MEM,OUT1 (CSR7)         ;REG. ADDRESS.
4748          000015                  MICPC=MICPC+1
4749  017206  041227                  .WORD   .$.
4750  017210                          CALL    ERROR                   ;REPORT DATA ERROR.
4751  017210                          MOVE    # (MICPC+3),BREG
4752          000016                  MICPC=MICPC+1
4753  017210  000420                  .WORD   .$.
4754  017212                          SBR     ERROR
4755          000017                  MICPC=MICPC+1
4756  017212  104400                  .WORD   .$.
4757  017214                          MOVE    SPAD (4),BREG           ;RESTORE BREG...
4758          000020                  MICPC=MICPC+1
4759  017214  060604                  .WORD   .$.
4760  017216                          SBR     2$                      ;LOOP ON ERROR...
4761          000021                  MICPC=MICPC+1
4762  017216  100403                  .WORD   .$.
4763  017220                  3$:     CALL    SCP1                    ;IS LOOP DATA SET.
```

```
4764  017220                    MOVE    # <MICPC+3>,BREG
4765          000022            MICPC=MICPC+1
4766  017220  000424            .WORD   .$.
4767  017222                    SBR     SCP1
4768          000023            MICPC=MICPC+1
4769  017222  104427            .WORD   .$.
4770  017224                    MOVE    SPAD <4>,BREG
4771          000024            MICPC=MICPC+1
4772  017224  060604            .WORD   .$.
4773  017226                    SBR     2$                      ;YES, DO IT.
4774          000025            MICPC=MICPC+1
4775  017226  100403            .WORD   .$.
4776  017230                    MOVE    SPAD <4>,BREG
4777          000026            MICPC=MICPC+1
4778  017230  060604            .WORD   .$.
4779  017232                    SHFBRT                          ;NO, CONTINUE...
4780          000027            MICPC=MICPC+1
4781  017232  061620            .WORD   .SBR!.SELB!.DBRSH
4782  017234                    BR7     4$                      ;IS IT DONE?...
4783          000030            MICPC=MICPC+1
4784  017234  103432            .WORD   .$.
4785  017236                    SBR     2$                      ;NO, CONTINUE...
4786          000031            MICPC=MICPC+1
4787  017236  100403            .WORD   .$.
4788  017240                4$:
4789  017240                SFLOT   OUT0,INP0,6,0,0,11$,12$,13$,14$
4790  017240                11$:  MOVE    # 177,BREG  ;START WITH BIT 7.
4791          000032            MICPC=MICPC+1
4792  017240  000577            .WORD   .$.
4793  017242                12$:
4794  017242                    MOVE    BREG,OUT0 <6>   ;SET THE BIT.
4795          000033            MICPC=MICPC+1
4796  017242  062226            .WORD   .$.
4797  017244                    MOVE    INP0 <6>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
4798          000034            MICPC=MICPC+1
4799  017244  023140            .WORD   .$.
4800  017246                    MOVE    BREG,SPAD <4>
4801          000035            MICPC=MICPC+1
4802  017246  063224            .WORD   .$.
4803  017250                    $IFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...
4804
4805
4806  017250                    SUB2C   SPAD <0>,BREG,NOP
4807          000036            MICPC=MICPC+1
4808  017250  060360            .WORD   .$.
4809  017252                    BZ      13$
4810          000037            MICPC=MICPC+1
4811  017252  101452            .WORD   .$.
4812  017254                    MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
4813          000040            MICPC=MICPC+1
4814  017254  061224            .WORD   .$.
4815  017256                    MOVE    INP0 <6>,OUT1 <CSR5>;BAD DATA...
4816          000041            MICPC=MICPC+1
4817  017256  021145            .WORD   .$.
4818  017260                    MOVE    # 1,MEM                 ;TYPE OF ERROR...
4819          000042            MICPC=MICPC+1
```

```
4820  017260  002401          .WORD   .$.
4821  017262                  MOVE    MEM,OUT1 <CSR3>         ;
                              MICPC=MICPC+1
4822          000043          .WORD   .$.
4823  017262  041223          MOVE    # 6,MEM         ;
4824  017264                  MICPC=MICPC+1
4825          000044          .WORD   .$.
4826  017264  002406          MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
4827  017266                  MICPC=MICPC+1
4828          000045          .WORD   .$.
4829  017266  041227          CALL    ERROR                   ;REPORT DATA ERROR.
4830  017270                  MOVE    # <MICPC+3>,BREG
4831  017270                  MICPC=MICPC+1
4832          000046          .WORD   .$.
4833  017270  000450          SBR     ERROR
4834  017272                  MICPC=MICPC+1
4835          000047          .WORD   .$.
4836  017272  104400          MOVE    SPAD <4>,BREG           ;RESTORE BREG...
4837  017274                  MICPC=MICPC+1
4838          000050          .WORD   .$.
4839  017274  060604          SBR     13$                     ;LOOP ON ERROR...
4840  017276                  MICPC=MICPC+1
4841          000051          .WORD   .$.
4842  017276  100433    13$:  CALL    SCP1                    ;IS LOOP DATA SET.
4843  017300                  MOVE    # <MICPC+3>,BREG
4844  017300                  MICPC=MICPC+1
4845          000052          .WORD   .$.
4846  017300  000454          SBR     SCP1
4847  017302                  MICPC=MICPC+1
4848          000053          .WORD   .$.
4849  017302  104427          MOVE    SPAD <4>,BREG
4850  017304                  MICPC=MICPC+1
4851          000054          .WORD   .$.
4852  017304  060604          SBR     12$                     ;YES, DO IT.
4853  017306                  MICPC=MICPC+1
4854          000055          .WORD   .$.
4855  017306  100433          MOVE    SPAD <4>,BREG
4856  017310                  MICPC=MICPC+1
4857          000056          .WORD   .$.
4858  017310  060604          SHFBRT                          ;NO, CONTINUE...
4859  017312                  MICPC=MICPC+1
4860          000057          .WORD   .SBR!.SELB!.DBRSH
4861  017312  061620          BBS     12$             ;
4862  017314                  MICPC=MICPC+1
4863          000060          .WORD   .$.
4864  017314  103433    14$:
4865  017316                  CALL    SCPE
4866  017316                  MOVE    # <MICPC+3>,BREG
4867  017316                  MICPC=MICPC+1
4868          000061          .WORD   .$.
4869  017316  000463          SBR     SCPE
4870  017320                  MICPC=MICPC+1
4871          000062          .WORD   .$.
4872  017320  104454          SBR     21$
4873  017322                  MICPC=MICPC+1
4874          000063          .WORD   .$.
4875  017322  100400
```

```
017324                          $FLT1   OUT0,INP0,7,0
017324                          $XZ


                                ;************************** TEST 16 **************************
                                ;* MICRO PROCESSOR OUT0 REGISTER WRITE/READ TEST.
                                ;* FLOAT A 1 THROUGH REGISTER OUT0 (7)
                                ;* FLOAT A 0THROUGH REGISTER OUT0 (7)
017324                          $XZ
                                ;:*************************************************************

017324                          $TSTN
                                ;  TEST 16
                                ;-------------
017324  012737  000016  001202  TST16:  MOV     #16,$TSTNM              ; LOAD THE NO. OF THIS TEST
017332  012737  017530  001442          MOV     #TST17,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
017340  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
017344  017360                          MCT16
017346  104022                          ERROR   22                      ;TIME OUT ERROR...
017350  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
017354  000177  162062                  JMP     @NEXT                   ;GO TO NEXT TEST...
017360                          MCT16:
017360                          21$:
017360                                  MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
000000                                  MICPC=MICPC+1
017360  000400                          .WORD   .$.
017362                                  MOVE    BREG,SPAD (16)  ;FOR RETURN ADDRESS PURPOSES...
000001                                  MICPC=MICPC+1
017362  063236                          .WORD   .$.
017364                          $FLOT   OUT0,INP0,7,0,1,1$,2$,3$,4$
017364                          1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
000002                                  MICPC=MICPC+1
017364  000600                          .WORD   .$.
017366                          2$:
017366                                  MOVE    BREG,OUT0 (7)   ;SET THE BIT.
000003                                  MICPC=MICPC+1
017366  062227                          .WORD   .$.
017370                                  MOVE    INP0 (7),SPAD (0)       ;GET THE "FOUND" IN SCRATCH PAD.
000004                                  MICPC=MICPC+1
017370  023160                          .WORD   .$.
017372                                  MOVE    BREG,SPAD (4)
000005                                  MICPC=MICPC+1
017372  063224                          .WORD   .$.
017374                                  $IFEQ   BREG,SPAD (0)  3$       ;CHECK THE DATA...


017374                                  SUB2C   SPAD (0),BREG,NOP
000006                                  MICPC=MICPC+1
017374  060360                          .WORD   .$.
017376                                  BZ      3$
000007                                  MICPC=MICPC+1
017376  101422                          .WORD   .$.
017400                                  MOVE    BREG,OUT1 (CSR4)        ;GOOD DATA...
000010                                  MICPC=MICPC+1
017400  061224                          .WORD   .$.
```

```
017402
                000011
017402  021165
017404
                000012
017404  002401
017406
                000013
017406  041223
017410
                000014
017410  002407
017412
                000015
017412  041227
017414
017414
                000016
017414  000420
017416
                000017
017416  104400
017420
                000020
017420  060604
017422
                000021
017422  100403
017424
017424
                000022
017424  000424
017426
                000023
017426  104427
017430
                000024
017430  060604
017432
                000025
017432  100403
017434
                000026
017434  060604
017436
                000027
017436  061620
017440
                000030
017440  103432
017442
                000031
017442  100403
017444
017444
017444
```

```
        MOVE    INPO (7),OUT1 (CSR5);BAD DATA...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 1,MEM                 ;TYPE OF ERROR...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,OUT1 (CSR3)         ;
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 7,MEM          ;
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,OUT1 (CSR7)         ;REG. ADDRESS.
        MICPC=MICPC+1
        .WORD   .$.
        CALL    EROR                    ;REPORT DATA ERROR.
        MOVE    # (MICPC+3),BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     EROR
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD (4),BREG           ;RESTORE BREG...
        MICPC=MICPC+1
        .WORD   .$.
        SBR     2$                      ;LOOP ON ERROR...
        MICPC=MICPC+1
        .WORD   .$.
3$:     CALL    SCP1                    ;IS LOOP DATA SET.
        MOVE    # (MICPC+3),BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     SCP1
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD (4),BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     2$                      ;YES, DO IT.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD (4),BREG
        MICPC=MICPC+1
        .WORD   .$.
        SHFBRT                          ;NO, CONTINUE...
        MICPC=MICPC+1
        .WORD   .SBR!.SELB!.DBRSH
        BR7     4$                      ;IS IT DONE?...
        MICPC=MICPC+1
        .WORD   .$.
        SBR     2$                      ;NO, CONTINUE...
        MICPC=MICPC+1
        .WORD   .$.
4$:
SFLOT   OUT0,INP0,7,0,0,11$,12$,13$,14$
11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
```

```
4988                000032          MICPC=MICPC+1
4989  017444        000577          .WORD   .$.
4990  017446                    12$:
4991  017446                        MOVE    BREG,OUT0 <7>   ;SET THE BIT.
4992                000033          MICPC=MICPC+1
4993  017446        062227          .WORD   .$.
4994  017450                        MOVE    INPO <7>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
4995                000034          MICPC=MICPC+1
4996  017450        023160          .WORD   .$.
4997  017452                        MOVE    BREG,SPAD <4>
4998                000035          MICPC=MICPC+1
4999  017452        063224          .WORD   .$.
5000  017454                        $IFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...
5001
5002
5003  017454                        SUB2C   SPAD <0>,BREG,NOP
5004                000036          MICPC=MICPC+1
5005  017454        060360          .WORD   .$.
5006  017456                        BZ      13$
5007                000037          MICPC=MICPC+1
5008  017456        101452          .WORD   .$.
5009  017460                        MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
5010                000040          MICPC=MICPC+1
5011  017460        061224          .WORD   .$.
5012  017462                        MOVE    INPO <7>,OUT1 <CSR5>;BAD DATA...
5013                000041          MICPC=MICPC+1
5014  017462        021165          .WORD   .$.
5015  017464                        MOVE    $ 1,MEM                 ;TYPE OF ERROR...
5016                000042          MICPC=MICPC+1
5017  017464        002401          .WORD   .$.
5018  017466                        MOVE    MEM,OUT1 <CSR3>         ;
5019                000043          MICPC=MICPC+1
5020  017466        041223          .WORD   .$.
5021  017470                        MOVE    $ 7,MEM         ;
5022                000044          MICPC=MICPC+1
5023  017470        002407          .WORD   .$.
5024  017472                        MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
5025                000045          MICPC=MICPC+1
5026  017472        041227          .WORD   .$.
5027  017474                        CALL    ERROR                   ;REPORT DATA ERROR.
5028  017474                        MOVE    @ <MICPC+3>,BREG
5029                000046          MICPC=MICPC+1
5030  017474        000450          .WORD   .$.
5031  017476                        SOR     ERROR
5032                000047          MICPC=MICPC+1
5033  017476        104400          .WORD   .$.
5034  017500                        MOVE    SPAD <4>,BREG           ;RESTORE BREG...
5035                000050          MICPC=MICPC+1
5036  017500        060604          .WORD   .$.
5037  017502                        SOR     12$                     ;LOOP ON ERROR...
5038                000051          MICPC=MICPC+1
5039  017502        100433          .WORD   .$.
5040  017504                    13$:CALL    SCP1                    ;IS LOOP DATA SET.
5041  017504                        MOVE    @ <MICPC+3>,BREG
5042                000052          MICPC=MICPC+1
5043  017504        000454          .WORD   .$.
```

```
017506                          SBR     SCP1
                                MICPC=MICPC+1
000053                          .WORD   .$.
017506  104427                  MOVE    SPAD (4),BREG
                                MICPC=MICPC+1
000054                          .WORD   .$.
017510  060604                  SBR     12$                     ;YES, DO IT.
                                MICPC=MICPC+1
000055                          .WORD   .$.
017512  100433                  MOVE    SPAD (4),BREG
                                MICPC=MICPC+1
000056                          .WORD   .$.
017514  060604                  SHFRT                           ;NO, CONTINUE...
                                MICPC=MICPC+1
000057                          .WORD   .SBR!.SELB!.DBRSH
017516  061620                  BR      12$                     ;
                                MICPC=MICPC+1
000060                          .WORD   .$.
017520  103433          14$:
                                CALL    SCPE
                                MOVE    # (MICPC+3),BREG
000061                          MICPC=MICPC+1
017522  000463                  .WORD   .$.
                                SBR     SCPE
000062                          MICPC=MICPC+1
017524  104454                  .WORD   .$.
                                SBR     21$
000063                          MICPC=MICPC+1
017526  100400                  .WORD   .$.
017530                  SPROC4
017530                  SXZ


                                ;************************** TEST 17 **************************
                                ;* MICRO PROCESSOR B REGISTER TEST
                                ;* FLOAT A 1 THROUGH THE BREG.
                                ;* FLOAT A 0 THROUGH THE BREG.
                                ;*
017530                  SXZ
                                ;;*********************************************************************
017530                  $TSTN
                                ;  TEST 17
                                ;-------------
017530  012737  000017  001202  TST17: MOV     #17,$TSTNM              ; LOAD THE NO. OF THIS TEST
017536  012737  017720  001442          MOV     #TST20,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
017544  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
017550  017564                  MCT17
017552  104022                  ERROR   22                      ;TIME OUT ERROR...
017554  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
017560  000177  161656                  JMP     @NEXT                   ;GO TO NEXT TEST...
017564                  MCT17:
017564                  21$:
017564                          MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16.
000000                          MICPC=MICPC+1
```

```
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  1$,1S,2S,3S,4S
5105  017570          1$:     MOVE    # 001,MEM               ;START WITH BIT0.
5106                          MICPC=MICPC+1
5107  017570  002401          .WORD   .$.
5108  017572                  MOVE    MEM,SPAD (2)            ;SET THE DATA IN SPAD0.
5109                          MICPC=MICPC+1
5110  017572  043222          .WORD   .$.
5111  017574          2S:     MOVE    MEM,BREG                ;SET THE BIT IN BREG.
5112                          MICPC=MICPC+1
5113  017574  040620          .WORD   .$.
5114  017576                  MOVE    BREG,SPAD (1)           ;PUT IT IN SPAD  (1).
5115                          MICPC=MICPC+1
5116  017576  063221          .WORD   .$.
5117  017600                  SIFEQ   MEM,SPAD (1)    3S      ;COMPARE DATA...
5118
5119
5120  017600                  SUB2C   SPAD (1),MEM,NOP
5121                          MICPC=MICPC+1
5122  017600  040361          .WORD   .$.
5123  017602                  BZ      4$
5124                          MICPC=MICPC+1
5125  017602  101416          .WORD   .$.
5126  017604                  MOVE    MEM,OUT1 (CSR4)         ;REPORT DATA ERROR.
5127                          MICPC=MICPC+1
5128  017604  041224          .WORD   .$.
5129  017606                  MOVE    BREG,OUT1 (CSR5)        ;  "    "     "   .
5130                          MICPC=MICPC+1
5131  017606  061225          .WORD   .$.
5132  017610                  MOVE    # 3,OUT1 (CSR3)         ;ERROR TYPE...
5133                          MICPC=MICPC+1
5134  017610  001003          .WORD   .$.
5135  017612                  CALL    ERROR1                  ;DATA ERROR...
5136  017612                  MOVE    # (MICPC+3),BREG
5137                          MICPC=MICPC+1
5138  017612  000415          .WORD   .$.
5139  017614                  SER     ERROR1
5140                          MICPC=MICPC+1
5141  017614  104401          .WORD   .$.
5142  017616                  SBR     2S                      ;LOOP ON ERROR...
5143                          MICPC=MICPC+1
5144  017616  100404          .WORD   .$.
5145  017620          3$:     CALL    SCP11                   ;LOOP THE DATA...
5146  017620                  MOVE    # (MICPC+3),BREG
5147                          MICPC=MICPC+1
5148  017620  000420          .WORD   .$.
5149  017622                  SBR     SCP11
5150                          MICPC=MICPC+1
5151  017622  104430          .WORD   .$.
5152  017624                  SBR     2S                      ;YES, LOOP...
5153                          MICPC=MICPC+1
5154  017624  100404          .WORD   .$.
5155  017626                  MOVE    MEM,BREG
```

```
                    MICPC=MICPC+1
                    .WORD   .$.
                    SHOL    SPAD (2)                ;SET THE NEXT BIT.
                    MICPC=MICPC+1
.WORD               .$.!.DSP
                    SHOC    SPAD (2)
                    MICPC=MICPC+1
.WORD               .$.!.DSP
                    MOVE    SPAD (2),MEM            ;
                    MICPC=MICPC+1
                    .WORD   .$.
                    BR7     4$                      ;DONE...
                    MICPC=MICPC+1
                    .WORD   .$.
                    SBR     2$                      ;NO, CONTINUE...
                    MICPC=MICPC+1
                    .WORD   .$.
4$:
SBRFLT  0,11$,12$,13$,14$
11$:    MOVE    # 376,MEM                           ;START WITH BIT0.
                    MICPC=MICPC+1
                    .WORD   .$.
                    MOVE    MEM,SPAD (2)            ;SET THE DATA IN SPAD0.
                    MICPC=MICPC+1
                    .WORD   .$.
12$:    MOVE    MEM,BREG                            ;SET THE BIT IN BREG.
                    MICPC=MICPC+1
                    .WORD   .$.
                    MOVE    BREG,SPAD (1)           ;PUT IT IN SPAD  (1).
                    MICPC=MICPC+1
                    .WORD   .$.
                    SIFEQ   MEM,SPAD (1)    13$     ;COMPARE DATA...

                    SUB2C   SPAD (1),MEM,NOP
                    MICPC=MICPC+1
                    .WORD   .$.
                    BZ      13$
                    MICPC=MICPC+1
                    .WORD   .$.
                    MOVE    MEM,OUT1 (CSR4)         ;REPORT DATA ERROR.
                    MICPC=MICPC+1
                    .WORD   .$.
                    MOVE    BREG,OUT1 (CSR5)        ;  "    "     "  .
                    MICPC=MICPC+1
                    .WORD   .$.
                    MOVE    # 3,OUT1 (CSR3)         ;ERROR TYPE...
                    MICPC=MICPC+1
                    .WORD   .$.
                    CALL    ERROR1                  ;DATA ERROR...
                    MOVE    # (MICPC+3),BREG
                    MICPC=MICPC+1
                    .WORD   .$.
                    SBR     ERROR1
                    MICPC=MICPC+1
                    .WORD   .$.
```

```
017670                          SOR     12$                     ;LOOP ON ERROR...
                        000042  MICPC=MICPC+1
017670  100431                  .WORD   .$.
017672                  13$:    CALL    SCP11                   ;LOOP THE DATA...
017672                          MOVE    @ (MICPC+3),BREG
                        000043  MICPC=MICPC+1
017672  000445                  .WORD   .$.
017674                          SOR     SCP11
                        000044  MICPC=MICPC+1
017674  104430                  .WORD   .$.
017676                          SOR     12$                     ;YES, LOOP...
                        000045  MICPC=MICPC+1
017676  100431                  .WORD   .$.
017700                          MOVE    MEM,BREG
                        000046  MICPC=MICPC+1
017700  040620                  .WORD   .$.
017702                          SADD    SPAD (2)                ;SET THE NEXT BIT.
                        000047  MICPC=MICPC+1
017702  063142          .WORD   .$.!.DSP
017704                          SADC    SPAD (2)
                        000050  MICPC=MICPC+1
017704  063102          .WORD   .$.!.DSP
017706                          MOVE    SPAD (2),MEM            ;
                        000051  MICPC=MICPC+1
017706  062602                  .WORD   .$.
017710                  88?     12$             ;CONTINUE, IF NOT DONE...
                        000052  MICPC=MICPC+1
017710  103431                  .WORD   .$.
017712                  14$:
017712                          CALL    SCPE
017712                          MOVE    @ (MICPC+3),BREG
                        000053  MICPC=MICPC+1
017712  000455                  .WORD   .$.
017714                          SOR     SCPE
                        000054  MICPC=MICPC+1
017714  104454                  .WORD   .$.
017716                          SOR     21$
                        000055  MICPC=MICPC+1
017716  100400                  .WORD   .$.
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
```

```
                                        JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
                                        MCT20
                                        ERROR   22                      ;TIME OUT ERROR...
                                        MOV     @STACK,SP               ;RESET STACK...
                                        JMP     @NEXT                   ;GO TO NEXT TEST...
                                MCT20:
                                21$:
                                        MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16...
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                $SPFLT  1,1$,2$,3$,4$,4
                                1$:     MOVE # 200,BREG ;START WITH BIT 7.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                2$:     MOVE    BREG,SPAD <4>   ;LOAD THE SCRATCH PAD...
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    SPAD <4>,MEM    ;GET THE "FOUND"...
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        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    # 4,BREG                ;LOAD ADDRESS.
                                        MICPC=MICPC+1
                                        .WORD   .$.
```

```
5324  020012                    MOVE    BREG,OUT1 <CSR7>        ;
                 000017         MICPC=MICPC+1
5326  020012     061227         .WORD   .$.
5327  020014                    CALL    ERROR                   ;DATA ERROR!!
5328  020014                    MOVE    # <MICPC+3>,BREG
                 000020         MICPC=MICPC+1
5330  020014     000422         .WORD   .$.
5331  020016                    SBR     ERROR
                 000021         MICPC=MICPC+1
5333  020016     104400         .WORD   .$.
5334  020020                    MOVE    MEM,BREG                ;RESTORE BREG...
                 000022         MICPC=MICPC+1
5336  020020     040620         .WORD   .$.
5337  020022                    SBR     2$                      ;LOOP ON ERROR...
                 000023         MICPC=MICPC+1
5339  020022     100403         .WORD   .$.
5340  020024              3$:   CALL    SCP1                    ;IS LOOP DATA SET???
5341  020024                    MOVE    # <MICPC+3>,BREG
                 000024         MICPC=MICPC+1
5343  020024     000426         .WORD   .$.
5344  020026                    SBR     SCP1
                 000025         MICPC=MICPC+1
5346  020026     104427         .WORD   .$.
5347  020030                    MOVE    MEM,BREG                ;RESTORE BREG...
                 000026         MICPC=MICPC+1
5349  020030     040620         .WORD   .$.
5350  020032                    SBR     2$                      ;LOOP ON DATA.
                 000027         MICPC=MICPC+1
5352  020032     100403         .WORD   .$.
5353  020034                    MOVE    MEM,BREG                ;RESTORE BREG...
                 000030         MICPC=MICPC+1
5355  020034     040620         .WORD   .$.
5356  020036                    SHFBRT                          ;SET THE NEXT BIT...
                 000031         MICPC=MICPC+1
5358  020036     061620         .WORD   .SBR!.SELB!.DBRSH
5359  020040                    BR7     4$                      ;BRANCH IF DONE...
                 000032         MICPC=MICPC+1
5361  020040     103434         .WORD   .$.
5362  020042                    SBR     2$                      ;CONTINUE...
                 000033         MICPC=MICPC+1
5364  020042     100403         .WORD   .$.
5365  020044              4$:
5366  020044              $SPFLT  0,11$,12$,13$,14$,4
5367  020044              11$:    MOVE # 177,BREG ;START WITH BIT 7.
                 000034         MICPC=MICPC+1
5369  020044     000577         .WORD   .$.
5370  020046              12$:    MOVE    BREG,SPAD <4>   ;LOAD THE SCRATCH PAD...
                 000035         MICPC=MICPC+1
5372  020046     063224         .WORD   .$.
5373  020050                    MOVE    SPAD <4>,MEM    ;GET THE "FOUND"...
                 000036         MICPC=MICPC+1
5375  020050     062604         .WORD   .$.
5376  020052                    MOVE    MEM,SPAD <0>            ;
                 000037         MICPC=MICPC+1
5378  020052     043220         .WORD   .$.
5379  020054                    MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
```

```
020054  000040          MICPC=MICPC+1
020054  062620          .WORD   .$.
020056                  $IFEQ   BREG,SPAD <0>   13$      ;IF GOOD.. CONTINUE...

020056                  SUB2C   SPAD <0>,BREG,NOP
                        MICPC=MICPC+1
020056  000041          .WORD   .$.
020060  060360          BZ      13$
020060                  MICPC=MICPC+1
020060  000042          .WORD   .$.
020062  101456          MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
020062                  MICPC=MICPC+1
020062  000043          .WORD   .$.
020064  061224          MOVE    SPAD.<0>,BREG           ;
020064                  MICPC=MICPC+1
020064  000044          .WORD   .$.
020066  060600          MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
020066                  MICPC=MICPC+1
020066  000045          .WORD   .$.
020070  061225          MOVE    #4,BREG                 ;TYPE OF ERROR...
020070                  MICPC=MICPC+1
020070  000046          .WORD   .$.
020072  000404          MOVE    BREG,OUT1 <CSR3>        ;
020072                  MICPC=MICPC+1
020072  000047          .WORD   .$.
020074  061223          MOVE    #4,BREG                 ;LOAD ADDRESS.
020074                  MICPC=MICPC+1
020074  000050          .WORD   .$.
020076  000404          MOVE    BREG,OUT1 <CSR7>        ;
020076                  MICPC=MICPC+1
020076  000051          .WORD   .$.
020100  061227          CALL    ERROR                   ;DATA ERROR!!
020100                  MOVE    # <MICPC+3>,BREG
020100  000052          MICPC=MICPC+1
020102  000454          .WORD   .$.
020102                  $BR     ERROR
020102  000053          MICPC=MICPC+1
020104  104400          .WORD   .$.
020104                  MOVE    MEM,BREG                ;RESTORE BREG...
020104  000054          MICPC=MICPC+1
020106  040620          .WORD   .$.
020106                  $BR     12$                     ;LOOP ON ERROR...
020106  000055          MICPC=MICPC+1
020110  100435          .WORD   .$.
020110          13$:    CALL    SCP1                    ;IS LOOP DATA SET???
020110  000056          MOVE    # <MICPC+3>,BREG
020110  000460          MICPC=MICPC+1
020112  000057          .WORD   .$.
020112  104427          $BR     SCP1
020114  000060          MICPC=MICPC+1
020114  040620          .WORD   .$.
020116                  MOVE    MEM,BREG                ;RESTORE BREG...
                        MICPC=MICPC+1
                        .WORD   .$.
                        $BR     12$                     ;LOOP ON DATA.
```

```
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    MEM,BREG                ;RESTORE BREG...
                        MICPC=MICPC+1
                        .WORD   .$.
                        SHFRT                           ;SET THE NEXT BIT...
                        MICPC=MICPC+1
                        .WORD   .SBR!.SELB!.DBRSH
                        BR7     12$                     ;BRANCH IF NOT DONE...
                        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
                        .WORD   .$.
                $SPTS1  5
                $XZ


                        ;************************** TEST 21 **************************
                        ;* SCRATCH PAD TEST FOR SP5
                        ;* FLOAT A 1 THROUGH SCRATCH PAD 5
                        ;* FLOAT A 0 THROUGH SCRATCH PAD 5
                $XZ
                        ;;*********************************************************************
                $TSTN
                        ;  TEST 21
                        ;-------------
020134 012737 000021 001202     TST21:  MOV     #21,$TSTNM              ; LOAD THE NO. OF THIS TEST
020142 012737 020350 001442             MOV     #TST22,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
020150 004737 035536                    JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
020154 020170                           MCT21
020156 104022                           ERROR   22                      ;TIME OUT ERROR...
020160 012706 001200                    MOV     #STACK,SP               ;RESET STACK...
020164 000177 161252                    JMP     @NEXT                   ;GO TO NEXT TEST...
020170                          MCT21:
020170                          21$:
020170                                  MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16...
                                        MICPC=MICPC+1
020170 000400                           .WORD   .$.
020172                                  MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES.
                                        MICPC=MICPC+1
020172 063236                           .WORD   .$.
020174                          $SPFLT  1,1$,2$,3$,4$,5
020174                          1$:     MOVE # 200,BREG ;START WITH BIT 7.
                                        MICPC=MICPC+1
020174 000600                           .WORD   .$.
```

```
5492  020176                  2$:   MOVE    BREG,SPAD <5>   ;LOAD THE SCRATCH PAD...
5493                 000003          MICPC=MICPC+1
5494  020176 063225                  .WORD   .$.
5495  020200                         MOVE    SPAD <5>,MEM    ;GET THE "FOUND"...
5496                 000004          MICPC=MICPC+1
5497  020200 062605                  .WORD   .$.
5498  020202                         MOVE    MEM,SPAD <0>            ;
5499                 000005          MICPC=MICPC+1
5500  020202 043220                  .WORD   .$.
5501  020204                         MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
5502                 000006          MICPC=MICPC+1
5503  020204 062620                  .WORD   .$.
5504  020206                         $IFEQ   BREG,SPAD <0>   3$      ;IF GOOD.. CONTINUE...
5505
5506
5507  020206                         SUB2C   SPAD <0>,BREG,NOP
5508                 000007          MICPC=MICPC+1
5509  020206 060360                  .WORD   .$.
5510  020210                         BZ      3$
5511                 000010          MICPC=MICPC+1
5512  020210 101424                  .WORD   .$.
5513  020212                         MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
5514                 000011          MICPC=MICPC+1
5515  020212 061224                  .WORD   .$.
5516  020214                         MOVE    SPAD <0>,BREG           ;
5517                 000012          MICPC=MICPC+1
5518  020214 060600                  .WORD   .$.
5519  020216                         MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
5520                 000013          MICPC=MICPC+1
5521  020216 061225                  .WORD   .$.
5522  020220                         MOVE    # 4,BREG                ;TYPE OF ERROR...
5523                 000014          MICPC=MICPC+1
5524  020220 000404                  .WORD   .$.
5525  020222                         MOVE    BREG,OUT1 <CSR3>        ;
5526                 000015          MICPC=MICPC+1
5527  020222 061223                  .WORD   .$.
5528  020224                         MOVE    # 5,BREG                ;LOAD ADDRESS.
5529                 000016          MICPC=MICPC+1
5530  020224 000405                  .WORD   .$.
5531  020226                         MOVE    BREG,OUT1 <CSR7>        ;
5532                 000017          MICPC=MICPC+1
5533  020226 061227                  .WORD   .$.
5534  020230                         CALL    ERROR                   ;DATA ERROR!!
5535  020230                         MOVE    # <MICPC+3>,BREG
5536                 000020          MICPC=MICPC+1
5537  020230 000422                  .WORD   .$.
5538  020232                         SBR     ERROR
5539                 000021          MICPC=MICPC+1
5540  020232 104400                  .WORD   .$.
5541  020234                         MOVE    MEM,BREG                ;RESTORE BREG...
5542                 000022          MICPC=MICPC+1
5543  020234 040620                  .WORD   .$.
5544  020236                         SBR     2$                      ;LOOP ON ERROR...
5545                 000023          MICPC=MICPC+1
5546  020236 100403                  .WORD   .$.
5547  020240                  3$:    CALL    S[illegible]1           ;IS LOOP DATA SET???
```

```
020240                          MOVE    @(MICPC+3),BREG
                                MICPC=MICPC+1
020240  000054                  .WORD   .S.
                                SBR     SCP1
020242                          MICPC=MICPC+1
020242  104427                  .WORD   .S.
                                MOVE    MEM,BREG                ;RESTORE BREG...
020244                          MICPC=MICPC+1
020244  000026                  .WORD   .S.
                                SBR     2$                      ;LOOP ON DATA.
020246  040620                  MICPC=MICPC+1
020246                          .WORD   .S.
                                MOVE    MEM,BREG                ;RESTORE BREG...
020250  000027                  MICPC=MICPC+1
020250  100403                  .WORD   .S.
                                SHFRT                           ;SET THE NEXT BIT...
020252                          MICPC=MICPC+1
020252  000030                  .WORD   .SBR!.SELB!.DBRSH
020252  040620                  BR7     4$                      ;BRANCH IF DONE...
020254                          MICPC=MICPC+1
020254  000031                  .WORD   .S.
                                SBR     2$                      ;CONTINUE...
020256  061620                  MICPC=MICPC+1
020256                          .WORD   .S.
020260  000032          4$:
020260  103434          SSPFLT  0,11$,12$,13$,14$,5
020262                  11$:    MOVE @ 177,BREG ;START WITH BIT 7.
020262  000033                  MICPC=MICPC+1
020264  100403                  .WORD   .S.
020264                  12$:    MOVE    BREG,SPAD (5)   ;LOAD THE SCRATCH PAD...
020264                          MICPC=MICPC+1
020264                          .WORD   .S.
020266                          MOVE    SPAD (5),MEM    ;GET THE "FOUND"...
020266                          MICPC=MICPC+1
020266                          .WORD   .S.
020270                          MOVE    MEM,SPAD (0)            ;
020270                          MICPC=MICPC+1
020272                          .WORD   .S.
020272                          MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
020272                          MICPC=MICPC+1
020272                          .WORD   .S.
020274                          $IFEQ   BREG,SPAD (0)   13$     ;IF GOOD.. CONTINUE...
020274
020276
020276                          SUB2C   SPAD (0),BREG,NOP
020276                          MICPC=MICPC+1
020300                          .WORD   .S.
020300                          BZ      13$
020300                          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.
```

```
5604  020302
                000045
5606  020302    061225
5607  020304
                000046
5608  020304    000404
5610  020306
                000047
5611  020306    061223
5612  020310
                000050
5614  020310    000405
5615  020312
                000051
5617  020312    061227
5618  020314
                000052
5620  020314    000454
5622  020316
                000053
5624  020316    104400
5626  020320
                000054
5627  020320    040620
5630  020322
                000055
5631  020322    100435
5632  020324
                000056
5634  020324    000460
5636  020326
                000057
5637  020326    104427
5640  020330
                000060
5641  020330    040620
5642  020332
                000061
5644  020332    100435
5645  020334
                000062
5647  020334    040620
5650  020336
                000063
5652  020336    061620
5654  020340
                000064
5655  020340    103435
5656  020342
5657  020342
5660  020342
                000065
5661  020342    000467
5662  020344
```

```
        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   .$.
        CALL    ERROR                   ;DATA ERROR!!
        MOVE    # <MICPC+3>,BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     ERROR
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,BREG                ;RESTORE BREG...
        MICPC=MICPC+1
        .WORD   .$.
        SBR     12$                     ;LOOP ON ERROR...
        MICPC=MICPC+1
        .WORD   .$.
13$:    CALL    SCP1                    ;IS LOOP DATA SET???
        MOVE    # <MICPC+3>,BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     SCP1
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,BREG                ;RESTORE BREG...
        MICPC=MICPC+1
        .WORD   .$.
        SBR     12$                     ;LOOP ON DATA.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,BREG                ;RESTORE BREG...
        MICPC=MICPC+1
        .WORD   .$.
        SHFBRT                          ;SET THE NEXT BIT...
        MICPC=MICPC+1
        .WORD   .SBR!.SELB!.DBRSH
        BBT     12$             ;BRANCH IF NOT DONE...
        MICPC=MICPC+1
        .WORD   .$.
14$:
        CALL    SCPE
        MOVE    # <MICPC+3>,BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     SCPE
```

```
5660                000066                          MICPC=MICPC+1
5661   020344       104454                          .WORD   .$.
5662   020346                                       SBR     21$
5663                000067                          MICPC=MICPC+1
5664   020346       100400                          .WORD   .$.
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                           $TSTN
5677                                                ;  TEST 22
5678                                                ;-------------
5679   020350  012737  000022  001202   TST22:     MOV     #22,$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,LDVRWT               ;LOAD-VERIFY-WAIT.
5683   020370  020404                              MCT22
5684   020372  104022                              ERROR   22                      ;TIME OUT ERROR...
5685   020374  012706  001200                      MOV     #STACK,SP               ;RESET STACK...
5686   020400  000177  161036                      JMP     @NEXT                   ;GO TO NEXT TEST...
5687   020404                           MCT22:
5688   020404                           21$:
5689   020404                                       MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16...
5690                       000000                   MICPC=MICPC+1
5691   020404              000400                   .WORD   .$.
5692   020406                                       MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES.
5693                       000001                   MICPC=MICPC+1
5694   020406              063236                   .WORD   .$.
5695   020410                           $SPFLT  1,1$,2$,3$,4$,6
5696   020410                           1$:         MOVE    # 200,BREG ;START WITH BIT 7.
5697                       000002                   MICPC=MICPC+1
5698   020410              000600                   .WORD   .$.
5699   020412                           2$:         MOVE    BREG,SPAD <6>   ;LOAD THE SCRATCH PAD...
5700                       000003                   MICPC=MICPC+1
5701   020412              063226                   .WORD   .$.
5702   020414                                       MOVE    SPAD <6>,MEM    ;GET THE "FOUND"...
5703                       000004                   MICPC=MICPC+1
5704   020414              062606                   .WORD   .$.
5705   020416                                       MOVE    MEM,SPAD <0>            ;
5706                       000005                   MICPC=MICPC+1
5707   020416              043220                   .WORD   .$.
5708   020420                                       MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
5709                       000006                   MICPC=MICPC+1
5710   020420              062620                   .WORD   .$.
5711   020422                                       SIFEQ   BREG,SPAD <0>   3$      ;IF GOOD.. CONTINUE...
5712
5713
5714   020422                                       SUB2C   SPAD <0>,BREG,NOP
5715                       000007                   MICPC=MICPC+1
```

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

```
                        .WORD   .SBR!.SELB!.DBRSH
                        BR      4$                      ;BRANCH IF DONE...
                        MICPC=MICPC+1
                        .WORD   .$.
                        BR      2$                      ;CONTINUE...
                        MICPC=MICPC+1
                        .WORD   .$.
4$:
SBRFLT  0,11$,12$,13$,14$,6
11$:    MOVE    # 177,BREG ;START WITH BIT 7.
                        MICPC=MICPC+1
                        .WORD   .$.
12$:    MOVE    BREG,SPAD <6>   ;LOAD THE SCRATCH PAD...
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    SPAD <6>,MEM    ;GET THE "FOUND"...
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    MEM,SPAD <0>            ;
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
                        MICPC=MICPC+1
                        .WORD   .$.
                        SIFEQ   BREG,SPAD <0>   13$     ;IF GOOD.. CONTINUE...

                        SUB2C   SPAD <0>,BREG,NOP
                        MICPC=MICPC+1
                        .WORD   .$.
                        BZ      1$
                        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    # 6,BREG                ;LOAD ADDRESS.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    BREG,OUT1 <CSR7>        ;
                        MICPC=MICPC+1
                        .WORD   .$.
                        CALL    ERROR                   ;DATA ERROR!!
                        MOVE    # <MICPC+3>,BREG
```

```
020520  000052          MICPC=MICPC+1
020522  000454          .WORD   .$.
020524                  SBR     ERROR
020526                  MICPC=MICPC+1
020530  000053          .WORD   .$.
020532  104400          MOVE    MEM,BREG                ;RESTORE BREG...
020534                  MICPC=MICPC+1
020536  000054          .WORD   .$.
020540  040620          SBR     12$                     ;LOOP ON ERROR...
020542                  MICPC=MICPC+1
020544  000055          .WORD   .$.
020546  100435  13$:    CALL    SCP1                    ;IS LOOP DATA SET???
020550                  MOVE    @ (MICPC+3),BREG
020552                  MICPC=MICPC+1
020554  000056          .WORD   .$.
020556  000460          SBR     SCP1
020560                  MICPC=MICPC+1
020562  000057          .WORD   .$.
020564  104427          MOVE    MEM,BREG                ;RESTORE BREG...
020566                  MICPC=MICPC+1
020570  000060          .WORD   .$.
020572  040620          SBR     12$                     ;LOOP ON DATA.
020574                  MICPC=MICPC+1
020576  000061          .WORD   .$.
020600  100435          MOVE    MEM,BREG                ;RESTORE BREG...
020602                  MICPC=MICPC+1
020604  000062          .WORD   .$.
020606  040620          SHFRT                           ;SET THE NEXT BIT...
020610                  MICPC=MICPC+1
020612  000063          .WORD   .SBR!.SELB!.DBRSH
020614  061620          BR7     12$             ;BRANCH IF NOT DONE...
020616                  MICPC=MICPC+1
020620  000064          .WORD   .$.
020622  103435  14$:
020624                  CALL    SCPE
020626                  MOVE    @ (MICPC+3),BREG
020630                  MICPC=MICPC+1
020632  000065          .WORD   .$.
020634  000467          SBR     SCPE
020636                  MICPC=MICPC+1
020640  000066          .WORD   .$.
020642  104454          SBR     21$
020644                  MICPC=MICPC+1
020646  000067          .WORD   .$.
020650  100400  $SPTS1  7
020652  020564  $XZ


                        ;*************************** TEST 23 ****************************
                        ;* SCRATCH PAD TEST FOR SP7
                        ;* FLOAT A 1 THROUGH SCRATCH PAD 7
                        ;* FLOAT A 0 THROUGH SCRATCH PAD 7
020654  020564  $XZ
                        ;:************************************************************

020656  020564  $TSTN
```

```
                                        ;  TEST 23
                                        ;--------------
020564 012737 000023 001202 TST23:  MOV     #23,$TSTNM              ; LOAD THE NO. OF THIS TEST
020572 012737 021000 001442         MOV     #TST24,NEXT             ; POINT TO THE START OF NEXT TEST.
                                    ;R1 CONTAINS BASE KMC11 ADDRESS
020600 004737 035536                JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
020604 020620                       MCT23
020606 104022                       ERROR   22                      ;TIME OUT ERROR...
020610 012706 001200                MOV     #STACK,SP               ;RESET STACK...
020614 000177 160622                JMP     @NEXT                   ;GO TO NEXT TEST...
020620                          MCT23:
020620                          21$:
020620                              MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16...
000000                              MICPC=MICPC+1
020620 000400                       .WORD   .$.
020622                              MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES.
000001                              MICPC=MICPC+1
020622 063236                       .WORD   .$.
020624                          $SPFLT  1,1$,2$,3$,4$,7
020624                          1$:     MOVE # 200,BREG ;START WITH BIT 7.
000002                              MICPC=MICPC+1
020624 000600                       .WORD   .$.
020626                          2$:     MOVE    BREG,SPAD <7>   ;LOAD THE SCRATCH PAD...
000003                              MICPC=MICPC+1
020626 063227                       .WORD   .$.
020630                              MOVE    SPAD <7>,MEM    ;GET THE "FOUND"...
000004                              MICPC=MICPC+1
020630 062607                       .WORD   .$.
020632                              MOVE    MEM,SPAD <0>            ;
000005                              MICPC=MICPC+1
020632 043220                       .WORD   .$.
020634                              MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
000006                              MICPC=MICPC+1
020634 062620                       .WORD   .$.
020636                              $IFEQ   BREG,SPAD <0>   3$      ;IF GOOD.. CONTINUE...


020636                              SUB2C   SPAD <0>,BREG,NOP
000007                              MICPC=MICPC+1
020636 060360                       .WORD   .$.
020640                              BZ      3$
000010                              MICPC=MICPC+1
020640 101424                       .WORD   .$.
020642                              MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
000011                              MICPC=MICPC+1
020642 061224                       .WORD   .$.
020644                              MOVE    SPAD <0>,BREG           ;
000012                              MICPC=MICPC+1
020644 060600                       .WORD   .$.
020646                              MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
000013                              MICPC=MICPC+1
020646 061225                       .WORD   .$.
020650                              MOVE    # 4,BREG                ;TYPE OF ERROR...
000014                              MICPC=MICPC+1
020650 000404                       .WORD   .$.
020652                              MOVE    BREG,OUT1 <CSR3>        ;
```

```
5940                000015                  MICPC=MICPC+1
5941  020652  061223                        .WORD   .$.
5942  020654                                MOVE    # 7,BREG                ;LOAD ADDRESS.
5943                000016                  MICPC=MICPC+1
5944  020654  000407                        .WORD   .$.
5945  020656                                MOVE    BREG,OUT1 (CSR7)        ;
5946                000017                  MICPC=MICPC+1
5947  020656  061227                        .WORD   .$.
5948  020660                                CALL    EROR                    ;DATA ERROR!!
5949  020660                                MOVE    # (MICPC+3),BREG
5950                000020                  MICPC=MICPC+1
5951  020660  000422                        .WORD   .$.
5952  020662                                SBR     EROR
5953                000021                  MICPC=MICPC+1
5954  020662  104400                        .WORD   .$.
5955  020664                                MOVE    MEM,BREG                ;RESTORE BREG...
5956                000022                  MICPC=MICPC+1
5957  020664  040620                        .WORD   .$.
5958  020666                                SBR     2$                      ;LOOP ON ERROR...
5959                000023                  MICPC=MICPC+1
5960  020666  100403                        .WORD   .$.
5961  020670                        3$:     CALL    SCP1                    ;IS LOOP DATA SET???
5962  020670                                MOVE    # (MICPC+3),BREG
5963                000024                  MICPC=MICPC+1
5964  020670  000426                        .WORD   .$.
5965  020672                                SBR     SCP1
5966                000025                  MICPC=MICPC+1
5967  020672  104427                        .WORD   .$.
5968  020674                                MOVE    MEM,BREG                ;RESTORE BREG...
5969                000026                  MICPC=MICPC+1
5970  020674  040620                        .WORD   .$.
5971  020676                                SBR     2$                      ;LOOP ON DATA.
5972                000027                  MICPC=MICPC+1
5973  020676  100403                        .WORD   .$.
5974  020700                                MOVE    MEM,BREG                ;RESTORE BREG...
5975                000030                  MICPC=MICPC+1
5976  020700  040620                        .WORD   .$.
5977  020702                                SHFBRT                          ;SET THE NEXT BIT...
5978                000031                  MICPC=MICPC+1
5979  020702  061620                        .WORD   .SBR!.SELB!.DBRSH
5980  020704                                BR7     4$                      ;BRANCH IF DONE...
5981                000032                  MICPC=MICPC+1
5982  020704  103434                        .WORD   .$.
5983  020706                                SBR     2$                      ;CONTINUE...
5984                000033                  MICPC=MICPC+1
5985  020706  100403                        .WORD   .$.
5986  020710                        4$:
5987  020710                        $SPFLT  0,11$,12$,13$,14$,7
5988  020710                        11$:    MOVE # 177,BREG ;START WITH BIT 7.
5989                000034                  MICPC=MICPC+1
5990  020710  000577                        .WORD   .$.
5991  020712                        12$:    MOVE    BREG,SPAD (7)   ;LOAD THE SCRATCH PAD...
5992                000035                  MICPC=MICPC+1
5993  020712  063227                        .WORD   .$.
5994  020714                                MOVE    SPAD (7),MEM    ;GET THE "FOUND"...
5995                000036                  MICPC=MICPC+1
```

```
5996  020714  062607          .WORD   .$.
5997  020716                  MOVE    MEM,SPAD <0>            ;
5998          000037          MICPC=MICPC+1
5999  020716  043220          .WORD   .$.
6000  020720                  MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
6001          000040          MICPC=MICPC+1
6002  020720  062620          .WORD   .$.
6003  020722                  SIFEQ   BREG,SPAD <0>   13$     ;IF GOOD.. CONTINUE...
6004
6005
6006  020722                  SUB2C   SPAD <0>,BREG,NOP
6007          000041          MICPC=MICPC+1
6008  020722  060360          .WORD   .$.
6009  020724                  BZ      13$
6010          000042          MICPC=MICPC+1
6011  020724  101456          .WORD   .$.
6012  020726                  MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
6013          000043          MICPC=MICPC+1
6014  020726  061224          .WORD   .$.
6015  020730                  MOVE    SPAD <0>,BREG           ;
6016          000044          MICPC=MICPC+1
6017  020730  060600          .WORD   .$.
6018  020732                  MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
6019          000045          MICPC=MICPC+1
6020  020732  061225          .WORD   .$.
6021  020734                  MOVE    # 4,BREG                ;TYPE OF ERROR...
6022          000046          MICPC=MICPC+1
6023  020734  000404          .WORD   .$.
6024  020736                  MOVE    BREG,OUT1 <CSR3>        ;
6025          000047          MICPC=MICPC+1
6026  020736  061223          .WORD   .$.
6027  020740                  MOVE    # 7,BREG                ;LOAD ADDRESS.
6028          000050          MICPC=MICPC+1
6029  020740  000407          .WORD   .$.
6030  020742                  MOVE    BREG,OUT1 <CSR7>        ;
6031          000051          MICPC=MICPC+1
6032  020742  061227          .WORD   .$.
6033  020744                  CALL    EROR                    ;DATA ERROR!!
6034  020744                  MOVE    # <MICPC+3>,BREG
6035          000052          MICPC=MICPC+1
6036  020744  000454          .WORD   .$.
6037  020746                  SBR     EROR
6038          000053          MICPC=MICPC+1
6039  020746  104400          .WORD   .$.
6040  020750                  MOVE    MEM,BREG                ;RESTORE BREG...
6041          000054          MICPC=MICPC+1
6042  020750  040620          .WORD   .$.
6043  020752                  SBR     13$                     ;LOOP ON ERROR...
6044          000055          MICPC=MICPC+1
6045  020752  100435          .WORD   .$.
6046  020754          13$:    CALL    SCP1                    ;IS LOOP DATA SET???
6047  020754                  MOVE    # <MICPC+3>,BREG
6048          000056          MICPC=MICPC+1
6049  020754  000460          .WORD   .$.
6050  020756                  SBR     SCP1
6051          000057          MICPC=MICPC+1
```

```
020756  104427                  .WORD   .$.
020760                          MOVE    MEM,BREG                ;RESTORE BREG...
        000060                  MICPC=MICPC+1
020760  040620                  .WORD   .$.
020762                          SBR     1$                      ;LOOP ON DATA.
        000061                  MICPC=MICPC+1
020762  100435                  .WORD   .$.
020764                          MOVE    MEM,BREG                ;RESTORE BREG...
        000062                  MICPC=MICPC+1
020764  040620                  .WORD   .$.
020766                          SHFBRT                          ;SET THE NEXT BIT...
        000063                  MICPC=MICPC+1
020766  061620                  .WORD   .SBR!.SELB!.DBRSH
020770                          BR7     1$                      ;BRANCH IF NOT DONE...
        000064                  MICPC=MICPC+1
020770  103435                  .WORD   .$.
020772                  14$:
020772                          CALL    SCPE
        000065                  MOVE    # <MICPC+3>,BREG
020772  000467                  MICPC=MICPC+1
020774                          .WORD   .$.
        000066                  SBR     SCPE
020774  104454                  MICPC=MICPC+1
020776                          .WORD   .$.
        000067                  SBR     21$
020776  100400                  MICPC=MICPC+1
021000                  $SPTS1  10
021000                  $XZ
```

```
                        ;************************** TEST 24 ****************************
                        ;* SCRATCH PAD TEST FOR SP10
                        ;* FLOAT A 1 THROUGH SCRATCH PAD 10
                        ;* FLOAT A 0 THROUGH SCRATCH PAD 10
021000                  $XZ
                        ;;*************************************************************
021000                  $TSTN
                        ;  TEST 24
                        ;-------------
021000  012737  000024  001202  TST24:  MOV     #24,$TSTNM              ; LOAD THE NO. OF THIS TEST
021006  012737  021214  001442          MOV     #TST2S,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
021014  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
021020  021034                          MCT24
021022  104022                          ERROR   22                      ;TIME OUT ERROR...
021024  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
021030  000177  160406                  JMP     @NEXT                   ;GO TO NEXT TEST...
021034                  MCT24:
021034                  21$:
021034                          MOVE    # 0,BREG              ;SET TO CLEAR SPAD 16...
        000000                  MICPC=MICPC+1
021034  000400                  .WORD   .$.
021036                          MOVE    BREG,SPAD <16>        ;FOR RETURN ADDRESS PURPOSES.
        000001                  MICPC=MICPC+1
```

```
021036 063236                   .WORD   .$.
021040                  $SPFLT  1,1$,2$,3$,4$,10
021040                  1$:     MOVE    # 200,BREG ;START WITH BIT 7.
000002                          MICPC=MICPC+1
021040 000600                   .WORD   .$.
021042                  2$:     MOVE    BREG,SPAD <10>  ;LOAD THE SCRATCH PAD...
000003                          MICPC=MICPC+1
021042 063230                   .WORD   .$.
021044                          MOVE    SPAD <10>,MEM   ;GET THE "FOUND"...
000004                          MICPC=MICPC+1
021044 062610                   .WORD   .$.
021046                          MOVE    MEM,SPAD <0>            ;
000005                          MICPC=MICPC+1
021046 043220                   .WORD   .$.
021050                          MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
000006                          MICPC=MICPC+1
021050 062620                   .WORD   .$.
021052                          $IFEQ   BREG,SPAD <0>   3$      ;IF GOOD.. CONTINUE...

021052                          SUB2C   SPAD <0>,BREG,NOP
000007                          MICPC=MICPC+1
021052 060360                   .WORD   .$.
021054                          BZ      3$
000010                          MICPC=MICPC+1
021054 101424                   .WORD   .$.
021056                          MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
000011                          MICPC=MICPC+1
021056 061224                   .WORD   .$.
021060                          MOVE    SPAD <0>,BREG           ;
000012                          MICPC=MICPC+1
021060 060600                   .WORD   .$.
021062                          MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
000013                          MICPC=MICPC+1
021062 061225                   .WORD   .$.
021064                          MOVE    # 4,BREG                ;TYPE OF ERROR...
000014                          MICPC=MICPC+1
021064 000404                   .WORD   .$.
021066                          MOVE    BREG,OUT1 <CSR3>        ;
000015                          MICPC=MICPC+1
021066 061223                   .WORD   .$.
021070                          MOVE    # 10,BREG               ;LOAD ADDRESS.
000016                          MICPC=MICPC+1
021070 000410                   .WORD   .$.
021072                          MOVE    BREG,OUT1 <CSR7>        ;
000017                          MICPC=MICPC+1
021072 061227                   .WORD   .$.
021074                          CALL    ERROR                   ;DATA ERROR!!
021074                          MOVE    # <MICPC+3>,BREG
000020                          MICPC=MICPC+1
021074 000422                   .WORD   .$.
021076                          SBR     ERROR
000021                          MICPC=MICPC+1
021076 104400                   .WORD   .$.
021100                          MOVE    MEM,BREG                ;RESTORE BREG...
000022                          MICPC=MICPC+1
```

```
6164  021100  040620                .WORD   .$.
6165  021102                        SBR     2$                      ;LOOP ON ERROR...
6166          000023                MICPC=MICPC+1
6167  021102  100403                .WORD   .$.
6168  021104                3$:     CALL    SCP1                    ;IS LOOP DATA SET???
6169  021104                        MOVE    @ <MICPC+3>,BREG
6170          000024                MICPC=MICPC+1
6171  021104  000426                .WORD   .$.
6172  021106                        SBR     SCP1
6173          000025                MICPC=MICPC+1
6174  021106  104427                .WORD   .$.
6175  021110                        MOVE    MEM,BREG                ;RESTORE BREG...
6176          000026                MICPC=MICPC+1
6177  021110  040620                .WORD   .$.
6178  021112                        SBR     2$                      ;LOOP ON DATA.
6179          000027                MICPC=MICPC+1
6180  021112  100403                .WORD   .$.
6181  021114                        MOVE    MEM,BREG                ;RESTORE BREG...
6182          000030                MICPC=MICPC+1
6183  021114  040620                .WORD   .$.
6184  021116                        SHFRT                           ;SET THE NEXT BIT...
6185          000031                MICPC=MICPC+1
6186  021116  061620                .WORD   .SBR!.SELB!.DBRSH
6187  021120                        BBZ     4$                      ;BRANCH IF DONE...
6188          000032                MICPC=MICPC+1
6189  021120  103434                .WORD   .$.
6190  021122                        SBR     2$                      ;CONTINUE...
6191          000033                MICPC=MICPC+1
6192  021122  100403                .WORD   .$.
6193  021124                4$:
6194  021124                SSPFLT  0,11$,12$,13$,14$,10
6195  021124                11$:    MOVE @ 177,BREG ;START WITH BIT 7.
6196          000034                MICPC=MICPC+1
6197  021124  000577                .WORD   .$.
6198  021126                12$:    MOVE    BREG,SPAD <10>  ;LOAD THE SCRATCH PAD...
6199          000035                MICPC=MICPC+1
6200  021126  063230                .WORD   .$.
6201  021130                        MOVE    SPAD <10>,MEM   ;GET THE "FOUND"...
6202          000036                MICPC=MICPC+1
6203  021130  062610                .WORD   .$.
6204  021132                        MOVE    MEM,SPAD <0>            ;
6205          000037                MICPC=MICPC+1
6206  021132  043220                .WORD   .$.
6207  021134                        MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
6208          000040                MICPC=MICPC+1
6209  021134  062620                .WORD   .$.
6210  021136                        $IFEQ   BREG,SPAD <0>   13$     ;IF GOOD.. CONTINUE...
6211
6212
6213  021136                        SUB2C   SPAD <0>,BREG,NOP
6214          000041                MICPC=MICPC+1
6215  021136  060360                .WORD   .$.
6216  021140                        BZ      13$
6217          000042                MICPC=MICPC+1
6218  021140  101456                .WORD   .$.
6219  021142                        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   .$.
                        MOVE    # 10,BREG               ;LOAD ADDRESS.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    BREG,OUT1 <CSR7>        ;
                        MICPC=MICPC+1
                        .WORD   .$.
                        CALL    ERDR                    ;DATA ERROR!!
                        MOVE    # <MICPC+3>,BREG
                        MICPC=MICPC+1
                        .WORD   .$.
                        SBR     ERDR
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    MEM,BREG                ;RESTORE BREG...
                        MICPC=MICPC+1
                        .WORD   .$.
                        SBR     12$                     ;LOOP ON ERROR...
                        MICPC=MICPC+1
                        .WORD   .$.
135$:                   CALL    SCP1                    ;IS LOOP DATA SET???
                        MOVE    # <MICPC+3>,BREG
                        MICPC=MICPC+1
                        .WORD   .$.
                        SBR     SCP1
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    MEM,BREG                ;RESTORE BREG...
                        MICPC=MICPC+1
                        .WORD   .$.
                        SBR     12$                     ;LOOP ON DATA.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    MEM,BREG                ;RESTORE BREG...
                        MICPC=MICPC+1
                        .WORD   .$.
                        SHFBRT                          ;SET THE NEXT BIT...
                        MICPC=MICPC+1
                        .WORD   .SBR!.SELB!.DBRSH
                        BB7     12$             ;BRANCH IF NOT DONE...
                        MICPC=MICPC+1
                        .WORD   .$.
145$:
```

```
021206                          CALL    SCPE
021206                          MOVE    @(MICPC+3),BREG
                    000065      MICPC=MICPC+1
021206  000467                  .WORD   .$.
021210                          SBR     SCPE
                    000066      MICPC=MICPC+1
021210  104454                  .WORD   .$.
021212                          SBR     21$
                    000067      MICPC=MICPC+1
021212  100400                  .WORD   .$.
021214                  $SPTS1  11
021214                  $XZ


                                ;*************************** TEST 25 ****************************
                                ;* SCRATCH PAD TEST FOR SP11
                                ;* FLOAT A 1 THROUGH SCRATCH PAD 11
                                ;* FLOAT A 0 THROUGH SCRATCH PAD 11
021214                  $XZ     ;:***************************************************************

021214                  $TSTN
                                ;  TEST 25
                                ;--------------
021214  012737  000025  001202  TST25:  MOV     #25,$TSTNM              ; LOAD THE NO. OF THIS TEST
021222  012737  021430  001442         MOV     #TST26,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
021230  004737  035536                 JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
021234                          MCT25
021234  104022                  ERROR   22                      ;TIME OUT ERROR...
021236  012706  001200          MOV     #STACK,SP               ;RESET STACK...
021242  000177  160172          JMP     @NEXT                   ;GO TO NEXT TEST...
                                MCT25:
                                21$:
021246                          MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16...
                    000000      MICPC=MICPC+1
021246  000400                  .WORD   .$.
021250                          MOVE    BREG,SPAD (16)  ;FOR RETURN ADDRESS PURPOSES.
                    000001      MICPC=MICPC+1
021252  063236                  .WORD   .$.
021254                  $SPFLT  1,1$,2$,3$,4$,11
021254                  1$:     MOVE # 200,BREG ;START WITH BIT 7.
                    000002      MICPC=MICPC+1
021254  000600                  .WORD   .$.
021256                  2$:     MOVE    BREG,SPAD (11)  ;LOAD THE SCRATCH PAD...
                    000003      MICPC=MICPC+1
021256  063231                  .WORD   .$.
021260                          MOVE    SPAD (11),MEM   ;GET THE "FOUND"...
                    000004      MICPC=MICPC+1
021260  062611                  .WORD   .$.
021262                          MOVE    MEM,SPAD (0)            ;
                    000005      MICPC=MICPC+1
021262  043220                  .WORD   .$.
021264                          MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
                    000006      MICPC=MICPC+1
021264  062620                  .WORD   .$.
```

```
6332  021266                        SIFEQ   BREG,SPAD (0)   3$      ;IF GOOD.. CONTINUE...
6333
6334
6335  021266                        SUB2C   SPAD (0),BREG,NOP
6336          000007                MICPC=MICPC+1
6337  021266  060360                .WORD   .$.
6338  021270                        BZ      3$
6339          000010                MICPC=MICPC+1
6340  021270  101424                .WORD   .$.
6341  021272                        MOVE    BREG,OUT1 (CSR4)        ;ELSE, REPORT ERROR...
6342          000011                MICPC=MICPC+1
6343  021272  061224                .WORD   .$.
6344  021274                        MOVE    SPAD (0),BREG           ;
6345          000012                MICPC=MICPC+1
6346  021274  060600                .WORD   .$.
6347  021276                        MOVE    BREG,OUT1 (CSR5)        ;BAD DATA...
6348          000013                MICPC=MICPC+1
6349  021276  061225                .WORD   .$.
6350  021300                        MOVE    #4,BREG                 ;TYPE OF ERROR...
6351          000014                MICPC=MICPC+1
6352  021300  000404                .WORD   .$.
6353  021302                        MOVE    BREG,OUT1 (CSR3)        ;
6354          000015                MICPC=MICPC+1
6355  021302  061223                .WORD   .$.
6356  021304                        MOVE    #11,BREG                ;LOAD ADDRESS.
6357          000016                MICPC=MICPC+1
6358  021304  000411                .WORD   .$.
6359  021306                        MOVE    BREG,OUT1 (CSR7)        ;
6360          000017                MICPC=MICPC+1
6361  021306  061227                .WORD   .$.
6362  021310                        CALL    ERROR                   ;DATA ERROR!!
6363  021310                        MOVE    # (MICPC+3),BREG
6364          000020                MICPC=MICPC+1
6365  021310  000422                .WORD   .$.
6366  021312                        SBR     ERROR
6367          000021                MICPC=MICPC+1
6368  021312  104400                .WORD   .$.
6369  021314                        MOVE    MEM,BREG                ;RESTORE BREG...
6370          000022                MICPC=MICPC+1
6371  021314  040620                .WORD   .$.
6372  021316                        SBR     2$                      ;LOOP ON ERROR...
6373          000023                MICPC=MICPC+1
6374  021316  100403                .WORD   .$.
3$:   021320                        CALL    SCP1                    ;IS LOOP DATA SET???
6376  021320                        MOVE    # (MICPC+3),BREG
6377          000024                MICPC=MICPC+1
6378  021320  000426                .WORD   .$.
6379  021322                        SBR     SCP1
6380          000025                MICPC=MICPC+1
6381  021322  104427                .WORD   .$.
6382  021324                        MOVE    MEM,BREG                ;RESTORE BREG...
6383          000026                MICPC=MICPC+1
6384  021324  040620                .WORD   .$.
6385  021326                        SBR     2$                      ;LOOP ON DATA.
6386          000027                MICPC=MICPC+1
6387  021326  100403                .WORD   .$.
```

```
6388  021330
6389                000030
6390  021330       040620
6391  021332
6392                000031
6393  021332       061620
6394  021334
6395                000032
6396  021334       103434
6397  021336
6398                000033
6399  021336       100403
6400  021340
6401  021340
6402  021340
6403                000034
6404  021340       000577
6405  021342
6406                000035
6407  021342       063231
6408  021344
6409                000036
6410  021344       062611
6411  021346
6412                000037
6413  021346       043220
6414  021350
6415                000040
6416  021350       062620
6417  021352
6418
6419
6420  021352
6421                000041
6422  021352       060360
6423  021354
6424                000042
6425  021354       101456
6426  021356
6427                000043
6428  021356       061224
6429  021360
6430                000044
6431  021360       060600
6432  021362
6433                000045
6434  021362       061225
6435  021364
6436                000046
6437  021364       000404
6438  021366
6439                000047
6440  021366       061223
6441  021370
6442                000050
6443  021370       000411
```

```
        MOVE    MEM,BREG                ;RESTORE BREG...
        MICPC=MICPC+1
        .WORD   .$.
        SHFBRT                          ;SET THE NEXT BIT...
        MICPC=MICPC+1
        .WORD   .SBR!.SELB!.DBRSH
        BBZ     4$                      ;BRANCH IF DONE...
        MICPC=MICPC+1
        .WORD   .$.
        SBR     2$                      ;CONTINUE...
        MICPC=MICPC+1
        .WORD   .$.
4$:
SSPFLT  0,11$,12$,13$,14$,11
11$:    MOVE  # 177,BREG ;START WITH BIT 7.
        MICPC=MICPC+1
        .WORD   .$.
12$:    MOVE    BREG,SPAD <11>  ;LOAD THE SCRATCH PAD...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD <11>,MEM   ;GET THE "FOUND"...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,SPAD <0>            ;
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
        MICPC=MICPC+1
        .WORD   .$.
        SIFEQ   BREG,SPAD <0>   13$     ;IF GOOD.. CONTINUE...

        SUB2C   SPAD <0>,BREG,NOP
        MICPC=MICPC+1
        .WORD   .$.
        BZ      13$
        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    # 11,BREG               ;LOAD ADDRESS.
        MICPC=MICPC+1
        .WORD   .$.
```

```
021372
                000051
021372          061227
021374
021374
                000052
021374          000454
021376
                000053
021376          104400
021400
                000054
021400          040620
021402
                000055
021402          100435
021404
021404
                000056
021404          000460
021406
                000057
021406          104427
021410
                000060
021410          040620
021412
                000061
021412          100435
021414
                000062
021414          040620
021416
                000063
021416          061620
021420
                000064
021420          100435
                000065
                000467
                000066
                104454
                000067
                100400
```

```
                MOVE    BREG,OUT1 (CSR7)        ;
                MICPC=MICPC+1
                .WORD   .$.
                CALL    ERROR                   ;DATA ERROR!!
                MOVE    # (MICPC+3),BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     ERROR
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,BREG                ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     12$                     ;LOOP ON ERROR...
                MICPC=MICPC+1
                .WORD   .$.
13$:            CALL    SCP1                    ;IS LOOP DATA SET???
                MOVE    # (MICPC+3),BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     SCP1
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,BREG                ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     12$                     ;LOOP ON DATA.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,BREG                ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$.
                SHFRT                           ;SET THE NEXT BIT...
                MICPC=MICPC+1
                .WORD   .SBR!.SELB!.DBRSH
                BR7     12$             ;BRANCH IF NOT DONE...
                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
                .WORD   .$.
$SPTS1  12
$XZ


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

```
                                        ;* FLOAT A 0 THROUGH SCRATCH PAD 12
021430                          $XZ
                                        ;:*************************************************************
021430                          $TSTN
                                ;  TEST 26
                                ;-------------
021430 012737 000026 001202     TST26:  MOV     #26,$TSTNM              ; LOAD THE NO. OF THIS TEST
021436 012737 021641 001442             MOV     #TST27,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
021444 004737 035536                    JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
021450 021464                           MCT26
021452 104022                           ERROR   22                      ;TIME OUT ERROR...
021454 012706 001200                    MOV     #STACK,SP               ;RESET STACK...
021460 000177 157756                    JMP     @NEXT                   ;GO TO NEXT TEST...
021464                          MCT26:
021464                          21$:
                                        MOVE    # 0,BREG         ;SET TO CLEAR SPAD 16...
000000                          MICPC=MICPC+1
021464 000400                           .WORD   .$.
021466                                  MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES.
000001                          MICPC=MICPC+1
021466 063236                           .WORD   .$.
021470                          $SPPFLT 1,1$,2$,3$,4$,12
021470                          1$:     MOVE # 200,BREG ;START WITH BIT 7.
000002                          MICPC=MICPC+1
021470 000600                           .WORD   .$.
021472                          2$:     MOVE    BREG,SPAD <12>  ;LOAD THE SCRATCH PAD...
000003                          MICPC=MICPC+1
021472 063232                           .WORD   .$.
021474                                  MOVE    SPAD <12>,MEM   ;GET THE "FOUND"...
000004                          MICPC=MICPC+1
021474 063612                           .WORD   .$.
021476                                  MOVE    MEM,SPAD <0>            ;
000005                          MICPC=MICPC+1
021476 043220                           .WORD   .$.
021500                                  MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
000006                          MICPC=MICPC+1
021500 063620                           .WORD   .$.
021502                                  $IFEQ   BREG,SPAD <0>   3$      ;IF GOOD.. CONTINUE...

021502                                  SUB2C   SPAD <0>,BREG,NOP
000007                          MICPC=MICPC+1
021502 060360                           .WORD   .$.
021504                                  BZ      3$
000010                          MICPC=MICPC+1
021504 101424                           .WORD   .$.
021506                                  MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
000011                          MICPC=MICPC+1
021506 061224                           .WORD   .$.
021510                                  MOVE    SPAD <0>,BREG           ;
000012                          MICPC=MICPC+1
021510 060600                           .WORD   .$.
021512                                  MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
000013                          MICPC=MICPC+1
```

```
                        .WORD   .$.
021512 061225           MOVE    # 4,BREG                ;TYPE OF ERROR...
021514                  MICPC=MICPC+1
000014                  .WORD   .$.
021514 000404           MOVE    BREG,OUT1 <CSR3>        ;
021516                  MICPC=MICPC+1
000015                  .WORD   .$.
021516 061223           MOVE    # 12,BREG               ;LOAD ADDRESS.
021520                  MICPC=MICPC+1
000016                  .WORD   .$.
021520 000412           MOVE    BREG,OUT1 <CSR7>        ;
021522                  MICPC=MICPC+1
000017                  .WORD   .$.
021522 061227           CALL    ERROR                   ;DATA ERROR!!
021524                  MOVE    # <MICPC+3>,BREG
000020                  MICPC=MICPC+1
021524 000422           .WORD   .$.
021526                  SOR     ERROR
000021                  MICPC=MICPC+1
021526 104400           .WORD   .$.
021530                  MOVE    MEM,BREG                ;RESTORE BREG...
000022                  MICPC=MICPC+1
021530 040620           .WORD   .$.
021532                  SOR     2$                      ;LOOP ON ERROR...
000023                  MICPC=MICPC+1
021532 100403           .WORD   .$.
021534          3$:     CALL    SCP1                    ;IS LOOP DATA SET???
000024                  MICPC=MICPC+1
021534 000426           .WORD   .$.
021536                  SOR     SCP1
000025                  MICPC=MICPC+1
021536 104427           .WORD   .$.
021540                  MOVE    MEM,BREG                ;RESTORE BREG...
000026                  MICPC=MICPC+1
021540 040620           .WORD   .$.
021542                  SOR     2$                      ;LOOP ON DATA.
000027                  MICPC=MICPC+1
021542 100403           .WORD   .$.
021544                  MOVE    MEM,BREG                ;RESTORE BREG...
000030                  MICPC=MICPC+1
021544 040620           .WORD   .$.
021546                  SHFBRT                          ;SET THE NEXT BIT...
000031                  MICPC=MICPC+1
021546 061620           .WORD   .SBR!.SELB!.DBRSH
021550                  BR7     4$                      ;BRANCH IF DONE...
000032                  MICPC=MICPC+1
021550 103434           .WORD   .$.
021552                  SOR     2$                      ;CONTINUE...
000033                  MICPC=MICPC+1
021552 100403           .WORD   .$.
021554          4$:
021554          SSPFLT  0,11$,12$,13$,14$,12
021554          11$:    MOVE # 177,BREG ;START WITH BIT 7.
000034                  MICPC=MICPC+1
021554 000577           .WORD   .$.
```

```
021556                  12$:    MOVE    BREG,SPAD (12)  ;LOAD THE SCRATCH PAD...
                                MICPC=MICPC+1
000035                          .WORD   .$.
021556  063232                  MOVE    SPAD (12),MEM   ;GET THE "FOUND"...
021560                          MICPC=MICPC+1
000036                          .WORD   .$.
021560  062612                  MOVE    MEM,SPAD (0)            ;
021562                          MICPC=MICPC+1
000037                          .WORD   .$.
021562  043220                  MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
021564                          MICPC=MICPC+1
000040                          .WORD   .$.
021564  062620                  SIFEQ   BREG,SPAD (0)   13$     ;IF GOOD.. CONTINUE...
021566

021566                          SUB2C   SPAD (0),BREG,NOP
000041                          MICPC=MICPC+1
021566  060360                  .WORD   .$.
021570                          BZ      13$
000042                          MICPC=MICPC+1
021570  101456                  .WORD   .$.
021572                          MOVE    BREG,OUT1 (CSR4)        ;ELSE, REPORT ERROR...
000043                          MICPC=MICPC+1
021572  061224                  .WORD   .$.
021574                          MOVE    SPAD (0),BREG           ;
000044                          MICPC=MICPC+1
021574  060600                  .WORD   .$.
021576                          MOVE    BREG,OUT1 (CSR5)        ;BAD DATA...
000045                          MICPC=MICPC+1
021576  061225                  .WORD   .$.
021600                          MOVE    #4,BREG                 ;TYPE OF ERROR...
000046                          MICPC=MICPC+1
021600  000404                  .WORD   .$.
021602                          MOVE    BREG,OUT1 (CSR3)        ;
000047                          MICPC=MICPC+1
021602  061223                  .WORD   .$.
021604                          MOVE    #12,BREG                ;LOAD ADDRESS.
000050                          MICPC=MICPC+1
021604  000412                  .WORD   .$.
021606                          MOVE    BREG,OUT1 (CSR7)        ;
000051                          MICPC=MICPC+1
021606  061227                  .WORD   .$.
021610                          CALL    ERROR                   ;DATA ERROR!!
021610                          MOVE    # (MICPC+3),BREG
000052                          MICPC=MICPC+1
021610  000454                  .WORD   .$.
021612                          SBR     ERROR
000053                          MICPC=MICPC+1
021612  104400                  .WORD   .$.
021614                          MOVE    MEM,BREG                ;RESTORE BREG...
000054                          MICPC=MICPC+1
021614  040620                  .WORD   .$.
021616                          SBR     12$                     ;LOOP ON ERROR...
000055                          MICPC=MICPC+1
021616  100435                  .WORD   .$.
021620                  13$:    CALL    SCP1                    ;IS LOOP DATA SET???
```

```
021620 000056
021620 000460          MOVE    #<MICPC+3>,BREG
                       MICPC=MICPC+1
                       .WORD   .$.
021622 000057          SBR     SCP1
021622 104427          MICPC=MICPC+1
                       .WORD   .$.
021624 000060          MOVE    MEM,BREG                ;RESTORE BREG...
021624 040620          MICPC=MICPC+1
021626 000061          .WORD   .$.
021626 100435          SBR     12$                     ;LOOP ON DATA.
021630 000062          MICPC=MICPC+1
021630 040620          .WORD   .$.
021632 000063          MOVE    MEM,BREG                ;RESTORE BREG...
021632 061620          MICPC=MICPC+1
021634 000064          .WORD   .$.
021634 103435          SHFRT                           ;SET THE NEXT BIT...
021636                 MICPC=MICPC+1
021636 000065          .WORD   .SBR!.SELB!.DBRSH
021636 000467          BR7     12$             ;BRANCH IF NOT DONE...
021640 000066          MICPC=MICPC+1
021640 104454          .WORD   .$.
021642 000067      14$:
021642 100400          CALL    SCPE
021644                 MOVE    #<MICPC+3>,BREG
021644                 MICPC=MICPC+1
                       .WORD   .$.
                       SBR     SCPE
                       MICPC=MICPC+1
                       .WORD   .$.
                       SBR     21$
                       MICPC=MICPC+1
                       .WORD   .$.
                SSPTS1 13
                       $XZ


                       ;*************************** TEST 27 ****************************
                       ;* SCRATCH PAD TEST FOR SP13
                       ;* FLOAT A 1 THROUGH SCRATCH PAD 13
                       ;* FLOAT A 0 THROUGH SCRATCH PAD 13
021644                 $XZ
                       ;:************************************************************
021644                 $TSTN
                       ;  TEST 27
                       ;--------------
021644 012737 000027 001202  TST27:  MOV     #27,$TSTNM              ; LOAD THE NO. OF THIS TEST
021652 012737 022060 001442          MOV     #TST30,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                             ;R1 CONTAINS BASE KMC11 ADDRESS
021660 004737 035536                 JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
021664 021700                        MCT27
021666 104022                        ERROR   22                      ;TIME OUT ERROR...
021670 012706 001200                 MOV     #STACK,SP               ;RESET STACK...
021674 000177 157542                 JMP     @NEXT                   ;GO TO NEXT TEST...
021700                 MCT27:
021700                 21$:
```

```
6724  021700
6725                000000
6726  021700       000400          MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16...
6727  021702                       MICPC=MICPC+1
6728                000001         .WORD   .$.
6729  021702       063236          MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES.
6730  021704                       MICPC=MICPC+1
6731  021704                       .WORD   .$.
6732                000002    $SPFLT  1,1$,2$,3$,4$,13
6733  021704       000600    1$:     MOVE    # 200,BREG ;START WITH BIT 7.
6734  021706                       MICPC=MICPC+1
6735                000003         .WORD   .$.
6736  021706       063233    2$:     MOVE    BREG,SPAD <13>  ;LOAD THE SCRATCH PAD...
6737  021710                       MICPC=MICPC+1
6738                000004         .WORD   .$.
6739  021710       062613          MOVE    SPAD <13>,MEM   ;GET THE "FOUND"...
6740  021712                       MICPC=MICPC+1
6741                000005         .WORD   .$.
6742  021712       043220          MOVE    MEM,SPAD <0>            ;
6743  021714                       MICPC=MICPC+1
6744                000006         .WORD   .$.
6745  021714       062620          MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
6746  021716                       MICPC=MICPC+1
6747                               .WORD   .$.
6748                               $IFEQ   BREG,SPAD <0>   3$      ;IF GOOD.. CONTINUE...
6749  021716
6750                000007         SUB2C   SPAD <0>,BREG,NOP
6751  021716       060360          MICPC=MICPC+1
6752  021720                       .WORD   .$.
6753                000010         BZ      3$
6754  021720       101424          MICPC=MICPC+1
6755  021722                       .WORD   .$.
6756                000011         MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
6757  021722       061224          MICPC=MICPC+1
6758  021724                       .WORD   .$.
6759                000012         MOVE    SPAD <0>,BREG           ;
6760  021724       060600          MICPC=MICPC+1
6761  021726                       .WORD   .$.
6762                000013         MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
6763  021730       061225          MICPC=MICPC+1
6764                               .WORD   .$.
6765                000014         MOVE    # 4,BREG                ;TYPE OF ERROR...
6766  021732       010404          MICPC=MICPC+1
6767  021734                       .WORD   .$.
6768                000015         MOVE    BREG,OUT1 <CSR3>        ;
6769  021734       061223          MICPC=MICPC+1
6770  021736                       .WORD   .$.
6771                000016         MOVE    # 13,BREG               ;LOAD ADDRESS.
6772  021736       000413          MICPC=MICPC+1
6773  021740                       .WORD   .$.
6774                000017         MOVE    BREG,OUT1 <CSR7>        ;
6775  021740       061227          MICPC=MICPC+1
6776  021740                       .WORD   .$.
6777  021740                       CALL    ERROR                   ;DATA ERROR!!
6778                000020         MOVE    # <MICPC+3>,BREG
6779  021740       000422          MICPC=MICPC+1
                                   .WORD   .$.
```

```
6780  021742                        SBR     ERROR
6781                000021          MICPC=MICPC+1
6782  021742  104400                .WORD   .$.
6783  021744                        MOVE    MEM,BREG                ;RESTORE BREG...
6784                000022          MICPC=MICPC+1
6785  021744  040620                .WORD   .$.
6786  021746                        SBR     2$                      ;LOOP ON ERROR...
6787                000023          MICPC=MICPC+1
6788  021746  100403                .WORD   .$.
6789  021750                3$:     CAL     SCP1                    ;IS LOOP DATA SET???
6790  021750                        MOVE    @ <MICPC+3),BREG
6791                000024          MICPC=MICPC+1
6792  021750  000426                .WORD   .$.
6793  021752                        SBR     SCP1
6794                000025          MICPC=MICPC+1
6795  021752  104427                .WORD   .$.
6796  021754                        MOVE    MEM,BREG                ;RESTORE BREG...
6797                000026          MICPC=MICPC+1
6798  021754  040620                .WORD   .$.
6799  021756                        SBR     2$                      ;LOOP ON DATA.
6800                000027          MICPC=MICPC+1
6801  021756  100403                .WORD   .$.
6802  021760                        MOVE    MEM,BREG                ;RESTORE BREG...
6803                000030          MICPC=MICPC+1
6804  021760  040620                .WORD   .$.
6805  021762                        SHFBRT                          ;SET THE NEXT BIT...
6806                000031          MICPC=MICPC+1
6807  021762  061620                .WORD   .SBR!.SELB!.DBRSH
6808  021764                        BB7     4$                      ;BRANCH IF DONE...
6809                000032          MICPC=MICPC+1
6810  021764  103434                .WORD   .$.
6811  021766                        SBR     2$                      ;CONTINUE...
6812                000033          MICPC=MICPC+1
6813  021766  100403                .WORD   .$.
6814  021770                4$:
6815  021770                SSPFLT  0,11$,12$,13$,14$,13
6816  021770                11$:    MOVE @ 177,BREG ;START WITH BIT 7.
6817                000034          MICPC=MICPC+1
6818  021770  000577                .WORD   .$.
6819  021772                12$:    MOVE    BREG,SPAD <13>  ;LOAD THE SCRATCH PAD...
6820                000035          MICPC=MICPC+1
6821  021772  063233                .WORD   .$.
6822  021774                        MOVE    SPAD <13>,MEM   ;GET THE "FOUND"...
6823                000036          MICPC=MICPC+1
6824  021774  062613                .WORD   .$.
6825  021776                        MOVE    MEM,SPAD <0>            ;
6826                000037          MICPC=MICPC+1
6827  021776  043220                .WORD   .$.
6828  022000                        MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
6829                000040          MICPC=MICPC+1
6830  022000  062620                .WORD   .$.
6831  022002                        SIFEQ   BREG,SPAD <0>   13$     ;IF GOOD.. CONTINUE...
6832
6833
6834  022002                        SUB2C   SPAD <0>,BREG,NOP
6835                000041          MICPC=MICPC+1
```

```
022002 060360            .WORD   .$
022004                   BZ      135
                         MICPC=MICPC+1
000042                   .WORD   .$.
022004 101456            MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
                         MICPC=MICPC+1
022006                   .WORD   .$.
000043                   MOVE    SPAD <0>,BREG           ;
022006 061224            MICPC=MICPC+1
022010                   .WORD   .$.
000044                   MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
022010 060600            MICPC=MICPC+1
022012                   .WORD   .$.
000045                   MOVE    # 4,BREG                ;TYPE OF ERROR...
022012 061225            MICPC=MICPC+1
022014                   .WORD   .$.
000046                   MOVE    BREG,OUT1 <CSR3>        ;
022014 000404            MICPC=MICPC+1
022016                   .WORD   .$.
000047                   MOVE    # 13,BREG               ;LOAD ADDRESS.
022016 061223            MICPC=MICPC+1
022020                   .WORD   .$.
000050                   MOVE    BREG,OUT1 <CSR7>        ;
022020 000413            MICPC=MICPC+1
022022                   .WORD   .$.
000051                   CALL    ERROR                   ;DATA ERROR!!
022022 061227            MOVE    # <MICPC+3>,BREG
022024                   MICPC=MICPC+1
022024                   .WORD   .$.
000052                   SBR     ERROR
022024 000454            MICPC=MICPC+1
022026                   .WORD   .$.
000053                   MOVE    MEM,BREG                ;RESTORE BREG...
022026 104400            MICPC=MICPC+1
022030                   .WORD   .$.
000054                   SBR     12$                     ;LOOP ON ERROR...
022030 040620            MICPC=MICPC+1
022032                   .WORD   .$.
000055           135:    CALL    SCP1                    ;IS LOOP DATA SET???
022032 100435            MOVE    # <MICPC+3>,BREG
022034                   MICPC=MICPC+1
022034                   .WORD   .$.
000056                   SBR     SCP1
022034 000460            MICPC=MICPC+1
022036                   .WORD   .$.
000057                   MOVE    MEM,BREG                ;RESTORE BREG...
022036 104427            MICPC=MICPC+1
022040                   .WORD   .$.
000060                   SBR     12$                     ;LOOP ON DATA.
022040 040620            MICPC=MICPC+1
022042                   .WORD   .$.
000061                   MOVE    MEM,BREG                ;RESTORE BREG...
022042 100435            MICPC=MICPC+1
022044                   .WORD   .$.
000062                   SHFRT                           ;SET THE NEXT BIT...
022044 040620            MICPC=MICPC+1
022046
000063
```

```
6892  022046  061620                    .WORD   .SBR!.SELB!.DBRSH
6893  022050                            BR7     12$             ;BRANCH IF NOT DONE...
6894          000064                    MICPC=MICPC+1
6895  022050  103435                    .WORD   .$.
6896  022052                    14$:
6897  022052                            CALL    SCPE
6898  022052                            MOVE    # <MICPC+3),BREG
6899          000065                    MICPC=MICPC+1
6900  022052  000467                    .WORD   .$.
6901  022054                            SBR     SCPE
6902          000066                    MICPC=MICPC+1
6903  022054  104454                    .WORD   .$.
6904  022056                            SBR     21$
6905          000067                    MICPC=MICPC+1
6906  022056  100400                    .WORD   .$.
6907  022060                    $SPTS1  14
6908  022060                    $XZ


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  022060                    $TSTN
6919                                    ;  TEST 30
6920                                    ;-------------
6921  022060  012737  000030  001202    TST30:  MOV     #30,$TSTNM              ; LOAD THE NO. OF THIS TEST
6922  022066  012737  022274  001442            MOV     #TST31,NEXT             ; POINT TO THE START OF NEXT TEST.
6923                                                                    ;R1 CONTAINS BASE KMC11 ADDRESS
6924  022074  004737  035536                    JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
6925  022100  022114                    MCT30
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                    21$:
6931  022114                            MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16...
6932          000000                    MICPC=MICPC+1
6933  022114  000400                    .WORD   .$.
6934  022116                            MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES.
6935          000001                    MICPC=MICPC+1
6936  022116  063236                    .WORD   .$.
6937  022120                    $SPFLT  1,1$,2$,3$,4$,14
6938  022120                    1$:     MOVE # 200,BREG ;START WITH BIT 7.
6939          000002                    MICPC=MICPC+1
6940  022120  000600                    .WORD   .$.
6941  022122                    2$:     MOVE    BREG,SPAD <14>  ;LOAD THE SCRATCH PAD...
6942          000003                    MICPC=MICPC+1
6943  022122  063234                    .WORD   .$.
6944  022124                            MOVE    SPAD <14>,MEM   ;GET THE "FOUND"...
6945          000004                    MICPC=MICPC+1
6946  022124  062614                    .WORD   .$.
6947  022126                            MOVE    MEM,SPAD <0>            ;
```

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

```
7004                    000026                  MICPC=MICPC+1
7005  022170  040620                            .WORD   .$.
7006  022172                                    SBR     2$                      ;LOOP ON DATA.
7007                    000027                  MICPC=MICPC+1
7008  022172  100403                            .WORD   .$.
7009  022174                                    MOVE    MEM,BREG                ;RESTORE BREG...
7010                    000030                  MICPC=MICPC+1
7011  022174  040620                            .WORD   .$.
7012  022176                                    SHFBRT                          ;SET THE NEXT BIT...
7013                    000031                  MICPC=MICPC+1
7014  022176  061620                            .WORD   .SBR!.SELB!.DBRSH
7015  022200                                    BR7     4$                      ;BRANCH IF DONE...
7016                    000032                  MICPC=MICPC+1
7017  022200  103434                            .WORD   .$.
7018  022202                                    SBR     2$                      ;CONTINUE...
7019                    000033                  MICPC=MICPC+1
7020  022202  100403                            .WORD   .$.
7021  022204                            4$:
7022  022204                            $SPFLT  0,11$,12$,13$,14$,14
7023  022204                            11$:    MOVE # 177,BREG ;START WITH BIT 7.
7024                    000034                  MICPC=MICPC+1
7025  022204  000577                            .WORD   .$.
7026  022206                            12$:    MOVE    BREG,SPAD (14)  ;LOAD THE SCRATCH PAD...
7027                    000035                  MICPC=MICPC+1
7028  022206  063234                            .WORD   .$.
7029  022210                                    MOVE    SPAD (14),MEM   ;GET THE "FOUND"...
7030                    000036                  MICPC=MICPC+1
7031  022210  062614                            .WORD   .$.
7032  022212                                    MOVE    MEM,SPAD (0)            ;
7033                    000037                  MICPC=MICPC+1
7034  022212  043220                            .WORD   .$.
7035  022214                                    MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
7036                    000040                  MICPC=MICPC+1
7037  022214  062620                            .WORD   .$.
7038  022216                                    $IFEQ   BREG,SPAD (0)   13$     ;IF GOOD.. CONTINUE...
7039
7040
7041  022216                                    SUB2C   SPAD (0),BREG,NOP
7042                    000041                  MICPC=MICPC+1
7043  022216  060360                            .WORD   .$.
7044  022220                                    BZ      13$
7045                    000042                  MICPC=MICPC+1
7046  022220  101456                            .WORD   .$.
7047  022222                                    MOVE    BREG,OUT1 (CSR4)        ;ELSE, REPORT ERROR...
7048                    000043                  MICPC=MICPC+1
7049  022222  061224                            .WORD   .$.
7050  022224                                    MOVE    SPAD (0),BREG           ;
7051                    000044                  MICPC=MICPC+1
7052  022224  060600                            .WORD   .$.
7053  022226                                    MOVE    BREG,OUT1 (CSR5)        ;BAD DATA...
7054                    000045                  MICPC=MICPC+1
7055  022226  061225                            .WORD   .$.
7056  022230                                    MOVE    # 4,BREG                ;TYPE OF ERROR...
7057                    000046                  MICPC=MICPC+1
7058  022230  000404                            .WORD   .$.
7059  022232                                    MOVE    BREG,OUT1 (CSR3)        ;
```

```
                        MICPC=MICPC+1
                        .WORD   .$
                        MOVE    #14,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     12$                     ;LOOP ON ERROR...
                        MICPC=MICPC+1
                        .WORD   .$
13$:                    CALL    SCP1                    ;IS LOOP DATA SET???
                        MOVE    # (MICPC+3),BREG
                        MICPC=MICPC+1
                        .WORD   .$
                        SBR     SCP1
                        MICPC=MICPC+1
                        .WORD   .$
                        MOVE    MEM,BREG                ;RESTORE BREG...
                        MICPC=MICPC+1
                        .WORD   .$
                        SBR     13$                     ;LOOP ON DATA.
                        MICPC=MICPC+1
                        .WORD   .$
                        MOVE    MEM,BREG                ;RESTORE BREG...
                        MICPC=MICPC+1
                        .WORD   .$
                        SHFBRT                          ;SET THE NEXT BIT...
                        MICPC=MICPC+1
                        .WORD   .SBR!.SELB!.DBRSH
                        BR7     13$             ;BRANCH IF NOT DONE...
                        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
                        .WORD   .$
$SPTS1  15
$XZ
```

```
                                        ;****************************** TEST 31 ****************************
                                        ;* SCRATCH PAD TEST FOR SP15
                                        ;* FLOAT A 1 THROUGH SCRATCH PAD 15
                                        ;* FLOAT A 0 THROUGH SCRATCH PAD 15
022274                          $XZ
                                        ;:*************************************************************************
022274                          $TSTN
                                ;  TEST 31
                                ;-------------
022274 012737 000031 001202 TST31:  MOV     #31,$TSTNM              ; LOAD THE NO. OF THIS TEST
022302 012737 022510 001442         MOV     #TST32,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                            ;R1 CONTAINS BASE KMC11 ADDRESS
022310 004737 035536                JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
022314 022330                       MCT31
022316 104022                       ERROR   22                      ;TIME OUT ERROR...
022320 012706 001200                MOV     #STACK,SP               ;RESET STACK...
022324 000177 157112                JMP     @NEXT                   ;GO TO NEXT TEST...
022330                          MCT31:
022330                          21$:
022330                              MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16...
000000                              MICPC=MICPC+1
022330 000400                       .WORD   .$.
022332                              MOVE    BREG,SPAD (16)  ;FOR RETURN ADDRESS PURPOSES.
000001                              MICPC=MICPC+1
022332 063236                       .WORD   .$.
022334                          SSPFLT  1,15,25,35,45,15
022334                          1$:     MOVE # 200,BREG ;START WITH BIT 7.
000002                              MICPC=MICPC+1
022334 000600                       .WORD   .$.
022336                          2$:     MOVE    BREG,SPAD (15)  ;LOAD THE SCRATCH PAD...
000003                              MICPC=MICPC+1
022336 063235                       .WORD   .$.
022340                              MOVE    SPAD (15),MEM   ;GET THE "FOUND"...
000004                              MICPC=MICPC+1
022340 062615                       .WORD   .$.
022342                              MOVE    MEM,SPAD (0)            ;
000005                              MICPC=MICPC+1
022342 043220                       .WORD   .$.
022344                              MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
000006                              MICPC=MICPC+1
022344 062620                       .WORD   .$.
022346                              $IFEQ   BREG,SPAD (0)   3$      ;IF GOOD.. CONTINUE...

022346                              SUB2C   SPAD (0),BREG,NOP
000007                              MICPC=MICPC+1
022346 060360                       .WORD   .$.
022350                              BZ      3$
000010                              MICPC=MICPC+1
022350 101424                       .WORD   .$.
022352                              MOVE    BREG,OUT1 (CSR4)        ;ELSE, REPORT ERROR...
000011                              MICPC=MICPC+1
022352 061224                       .WORD   .$.
```

```
022354
                000012
022356          060600
022360
                000012
022362          061225
022364
                000014
022366          000404
022370
                000015
022372          061223
022374
                000016
022376          000415
023000
                000017
023002          061227
023004
                000020
023006          000422
023010
                000021
023012          104400
023014
                000022
023016          040620
023020
                000023
023022          100403
023024
                000024
023026          000426
023030
                000025
023032          104427
023034
                000026
023036          040620
023040
                000027
023042          100403
023044
                000030
023046          040620
023050
                000031
023052          061620
023054
                000032
023056          103434
023060
                000033
023062          100403
```

```
        MOVE    SPRD (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    # 15,BREG               ;LOAD ADDRESS.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 (CSR7)        ;
        MICPC=MICPC+1
        .WORD   .$.
        CALL    ERROR                   ;DATA ERROR!!
        MOVE    # (MICPC+3),BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     ERROR
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,BREG                ;RESTORE BREG...
        MICPC=MICPC+1
        .WORD   .$.
        SBR     2$                      ;LOOP ON ERROR...
        MICPC=MICPC+1
        .WORD   .$.
3$:     CALL    SCP1                    ;IS LOOP DATA SET???
        MOVE    # (MICPC+3),BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     SCP1
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,BREG                ;RESTORE BREG...
        MICPC=MICPC+1
        .WORD   .$.
        SBR     2$                      ;LOOP ON DATA.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,BREG                ;RESTORE BREG...
        MICPC=MICPC+1
        .WORD   .$.
        SHFBRT                          ;SET THE NEXT BIT...
        MICPC=MICPC+1
        .WORD   .SBR!.SELB!.DRRSH
        BB7     4$                      ;BRANCH IF DONE...
        MICPC=MICPC+1
        .WORD   .$.
        SBR     2$                      ;CONTINUE...
        MICPC=MICPC+1
        .WORD   .$.
```

```
                                45:
                                SSPFLT  0,11$,12$,13$,14$,15
                        11$:    MOVE    # 177,BREG ;START WITH BIT 7.
                                MICPC=MICPC+1
                                .WORD   .$.
                        12$:    MOVE    BREG,SPAD <15>  ;LOAD THE SCRATCH PAD...
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    SPAD <15>,MEM   ;GET THE "FOUND"...
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    MEM,SPAD <0>            ;
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
                                MICPC=MICPC+1
                                .WORD   .$.
                                SIFEQ   BREG,SPAD <0>   13$     ;IF GOOD.. CONTINUE...


                                SUB2C   SPAD <0>,BREG,NOP
                                MICPC=MICPC+1
                                .WORD   .$.
                                BZ      13$
                                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    # 15,BREG               ;LOAD ADDRESS.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    BREG,OUT1 <CSR7>        ;
                                MICPC=MICPC+1
                                .WORD   .$.
                                CALL    ERROR                   ;DATA ERROR!!
                                MOVE    # <MICPC+3>,BREG
                                MICPC=MICPC+1
                                .WORD   .$.
                                SBR     ERROR
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    MEM,BREG                ;RESTORE BREG...
                                MICPC=MICPC+1
```

```
 7284  022460  040620                    .WORD   .$.
 7285  022462                            $BR     12$                     ;LOOP ON ERROR...
 7286          000055                    MICPC=MICPC+1
 7287  022462  100435                    .WORD   .$.
 7288  022464                    13$:    CALL    SCP1                    ;IS LOOP DATA SET???
 7289  022464                            MOVE    @ (MICPC+3),BREG
 7290          000056                    MICPC=MICPC+1
 7291  022464  000460                    .WORD   .$.
 7292  022466                            $BR     $$1
 7293          000057                    MICPC=MICPC+1
 7294  022466  104427                    .WORD   .$.
 7295  022470                            MOVE    MEM,BREG                ;RESTORE BREG...
 7296          000060                    MICPC=MICPC+1
 7297  022470  040620                    .WORD   .$.
 7298  022472                            $BR     12$                     ;LOOP ON DATA.
 7299          000061                    MICPC=MICPC+1
 7300  022472  100435                    .WORD   .$.
 7301  022474                            MOVE    MEM,BREG                ;RESTORE BREG...
 7302          000062                    MICPC=MICPC+1
 7303  022474  040620                    .WORD   .$.
 7304  022476                            SHFRT                           ;SET THE NEXT BIT...
 7305          000063                    MICPC=MICPC+1
 7306  022476  061620                    .WORD   .SBR!.SELB!.DBRSH
 7307  022500                            BR7     12$             ;BRANCH IF NOT DONE...
 7308          000064                    MICPC=MICPC+1
 7309  022500  103435                    .WORD   .$.
 7310  022502                    14$:
 7311  022502                            CALL    SCPE
 7312  022502                            MOVE    @ (MICPC+3),BREG
 7313          000065                    MICPC=MICPC+1
 7314  022502  000467                    .WORD   .$.
 7315  022504                            $BR     SCPE
 7316          000066                    MICPC=MICPC+1
 7317  022504  104454                    .WORD   .$.
 7318  022506                            $BR     21$
 7319          000067                    MICPC=MICPC+1
 7320  022506  100400                    .WORD   .$.
 7321  022510                    $NPR1
 7322  022510                    $XZ
 7323
 7324
 7325                                    ;*************************** TEST 32 ***************************
 7326                                    ;* NPR TEST
 7327                                    ;* TEST OF DATI, 1 WORD FROM 11 MEMORY TO UPROC
 7328  022510                    $XZ
 7329                                    ;;***********************************************************
 7330
 7331  022510                    $TSTN
 7332                                    ;  TEST 32
 7333                                    ;-------------
 7334  022510  012737  000032  001202  TST32:  MOV     #32,$TSTNM              ; LOAD THE NO. OF THIS TEST
 7335  022516  012737  022704  001442          MOV     #TST33,NEXT             ; POINT TO THE START OF NEXT TEST.
 7336                                                                  ;R1 CONTAINS BASE KMC11 ADDRESS
 7337  022524  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
 7338  022530  022544                          MCT32
 7339  022532  104022                          ERROR   22                      ;TIME OUT ERROR...
```

```
7340 022534 012706 001200          MCT32:  MOV     #STACK,SP               ;RESET STACK...
7341 022540 000177 156676                  JMP     @NEXT                   ;GO TO NEXT TEST...
7342 022544                        1$:     MOVE    # 0,BREG                ;SET TO CLEAR NPR IBUS REGISTERS.
7343 022544                                MICPC=MICPC+1
7344 022544 000000                         .WORD   .$.
7345 022546 000400                         MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS PURPOSE...
7346 022546                                MICPC=MICPC+1
7347 022550 000001                         .WORD   .$.
7348 022550 063236                         MOVE    BREG,OUT0 <0>           ;CLEAR IBUS+IN DATA+LB.
7349 022552                                MICPC=MICPC+1
7350 022552 000002                         .WORD   .$.
7351 022554 062220                         MOVE    BREG,OUT0 <1>           ;CLEAR IBUS+IN DATA+HB.
7352 022556                                MICPC=MICPC+1
7353 022556 000003                         .WORD   .$.
7354 022560 062221                         MOVE    BREG,OUT0 <4>           ;CLEAR INBA+LB.
7355 022562                                MICPC=MICPC+1
7356 022562 000004                         .WORD   .$.
7357 022564 062224                         MOVE    BREG,OUT0 <5>           ;CLEAR INBA+HB.
7358 022566                                MICPC=MICPC+1
7359 022566 000005                         .WORD   .$.
7360 022570 062225                         MOVE    BREG,OUT1 <CSR10>       ;CLEAR EA+BITS.
7361 022572                                MICPC=MICPC+1
7362 022572 000006                         .WORD   .$.
7363 022574 061230                         MOVE    # 125,BREG              ;SET GOOD DATA.
7364 022576                                MICPC=MICPC+1
7365 022576 000007                         .WORD   .$.
7366 022600 000525                         MOVE    BREG,SPAD <3>   ; " " " .
7367 022602                                MICPC=MICPC+1
7368 022602 000010                         .WORD   .$.
7369 022604 063223                         SHFRT
7370 022606                                MICPC=MICPC+1
7371 022606 000011                         .WORD   .SBR!.SELB!.DBRSH
7372 022610 061620                         MOVE    BREG,SPAD <4>           ; " " " .
7373 022612                                MICPC=MICPC+1
7374 022612 000012                         .WORD   .$.
7375 022614 063224                         MOVE    # <NPDAT&377>,BREG      ;SET ADDRESS.
7376 022616                                MICPC=MICPC+1
7377 022616 000013                         .WORD   .$.
7378 022620 000702                         MOVE    BREG,SPAD <1>           ; " " .
7379 022622                                MICPC=MICPC+1
7380 022622 000014                         .WORD   .$.
7381 022624 063221                         MOVE    # <NPDAT&177400/400>,BREG       ;
7382 022626                                MICPC=MICPC+1
7383 022626 000015                         .WORD   .$.
7384 022630 000445                         MOVE    BREG,SPAD <2>           ;
7385 022632                                MICPC=MICPC+1
7386 022632 000016                         .WORD   .$.
7387 022634 063222                 2$:     MOVE    SPAD <1>,BREG           ;SET INBA+ADDRESS
7388 022636                                MICPC=MICPC+1
7389 022636 000017                         .WORD   .$.
7390 022640 060601                         MOVE    BREG,OUT0 <4>           ;
7391 022642                                MICPC=MICPC+1
7392 022642 000020                         .WORD   .$.
7393 022644 062224                         MOVE    SPAD <2>,BREG           ;SET INBA+ADDRESS
7394 022646                                MICPC=MICPC+1
7395 022646 000021
```

```
7396  022606  060602                    .WORD   .$.
7397  022610                            MOVE    BREG,OUT0 (5)           ;
7398          000022                    MICPC=MICPC+1
7399  022610  062225                    .WORD   .$.
7400  022612                            MOVE    # 001,BREG              ;SET NPR BIT.
7401          000023                    MICPC=MICPC+1
7402  022612  000401                    .WORD   .$.
7403  022614                            MOVE    BREG,OUT1 (CSR10)               ;SET NPR RQ BIT WITH IN NPR & WRD.XFR
7404          000024                    MICPC=MICPC+1
7405  022614  061230                    .WORD   .$.
7406  022616                    3$:     MOVE    INP1 (CSR10),BREG               ;IS NPR DONE???
7407          000025                    MICPC=MICPC+1
7408  022616  120600                    .WORD   .$.
7409  022620                            BPO     3$                      ;NO, SPIN ON IT??
7410          000026                    MICPC=MICPC+1
7411  022620  102025                    .WORD   .$.
7412  022622                            MOVE    INP0 (0),BREG           ;GET THE DATA LB.
7413          000027                    MICPC=MICPC+
7414  022622  020400                    .WORD   .$.
7415  022624                            $IFEQ   BREG,SPAD (3)   4$      ;IF GOOD, CHECK NEXT BYTE.
7416
7417
7418  022624                            SUB2C   SPAD (3),BREG,NOP
7419          000030                    MICPC=MICPC+1
7420  022624  060363                    .WORD   .$.
7421  022626                            BZ      4$
7422          000031                    MICPC=MICPC+1
7423  022626  101442                    .WORD   .$.
7424  022630                            MOVE    BREG,OUT1 (CSR5)        ;MOVE GOOD DATA...
7425          000032                    MICPC=MICPC+1
7426  022630  061225                    .WORD   .$.
7427  022632                            MOVE    SPAD (3),BREG           ;MOVE BAD DATA...
7428          000033                    MICPC=MICPC+1
7429  022632  060603                    .WORD   .$.
7430  022634                            MOVE    BREG,OUT1 (CSR4)                ;
7431          000034                    MICPC=MICPC+1
7432  022634  061224                    .WORD   .$.
7433  022636                            MOVE    # 11,BREG               ;SET THE TYPE OF ERROR...
7434          000035                    MICPC=MICPC+1
7435  022636  000411                    .WORD   .$.
7436  022640                            MOVE    BREG,OUT1 (CSR3)        ;
7437          000036                    MICPC=MICPC+1
7438  022640  061223                    .WORD   .$.
7439  022642                            CALL    ERROR1                  ;REPORT ERROR...
7440  022642                            MOVE    @ (MICPC+3),BREG
7441          000037                    MICPC=MICPC+1
7442  022642  000441                    .WORD   .$.
7443  022644                            SBR     ERROR1
7444          000040                    MICPC=MICPC+1
7445  022644  104401                    .WORD   .$.
7446  022646                            SBR     1$                      ;
7447          000041                    MICPC=MICPC+1
7448  022646  100400                    .WORD   .$.
7449  022650                    4$:     MOVE    INP0 (1),BREG           ;GET THE NPR DATA MB.
7450          000042                    MICPC=MICPC+1
7451  022650  020420                    .WORD   .$.
```

```
022652                          SIFEQ   BREG,SPAD <4>   5$      ;IF GOOD, BRANCH.

022652                          SUB2C   SPAD <4>,BREG,NOP
000043                          MICPC=MICPC+1
022652  060364                  .WORD   .$.
022654                          BZ      5$
000044                          MICPC=MICPC+1
022654  101454                  .WORD   .$.
022656                          MOVE    BREG,OUT1 <CSR5>        ;GOOD DATA.
000045                          MICPC=MICPC+1
022656  061225                  .WORD   .$.
022660                          MOVE    SPAD <4>,OUT1 <CSR4>    ;BAD DATA.
000046                          MICPC=MICPC+1
022660  061204                  .WORD   .$.
022662                          MOVE    # 11,MEM                ;TYPE OF ERROR...
000047                          MICPC=MICPC+1
022662  002411                  .WORD   .$.
022664                          MOVE    MEM,OUT1 <CSR3>         ;
000050                          MICPC=MICPC+1
022664  041223                  .WORD   .$.
022666                          CALL    ERROR1                  ;DATA ERROR...
022666                          MOVE    @ <MICPC+3>,BREG
000051                          MICPC=MICPC+1
022666  000453                  .WORD   .$.
022670                          SBR     ERROR1
000052                          MICPC=MICPC+1
022670  104401                  .WORD   .$.
022672                          SBR     1$                      ;LOOP ON ERROR...
000053                          MICPC=MICPC+1
022672  100400                  .WORD   .$.
022674                  5$:     CALL    SCPE                    ;SCOPE THE TEST...
022674                          MOVE    # <MICPC+3>,BREG
000054                          MICPC=MICPC+1
022674  000456                  .WORD   .$.
022676                          SBR     SCPE
000055                          MICPC=MICPC+1
022676  104454                  .WORD   .$.
022700                          SBR     1$                      ;DO THE NEXT ITERATION...
000056                          MICPC=MICPC+1
022700  100400                  .WORD   .$.
022702    125   252     NPDAT:  .BYTE   125,252                 ;
022704                  $NPR2   21,WORD,1
022704                  $XZ


                                ;************************** TEST 33 **************************
                                ;* NPR TEST
                                ;* TEST OF DATO, 1 WORD FROM UPROC TO PDP11 MEMORY.
                                ;* THEN DATI OF THAT WORD AND CHECK....
022704                  $XZ
                                ;;*********************************************************************

022704                  $TSTN
                                ;  TEST 33
                                ;--------------
```

```
7508  022704  012737  000033  001202  TST33:  MOV     #33,$TSTNM              ; LOAD THE NO. OF THIS TEST
7509  022712  012737  023154  001442          MOV     #TST34,NEXT             ; POINT TO THE START OF NEXT TEST.
7510                                                          ;R1 CONTAINS BASE KMC11 ADDRESS
7511  022720  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
7512  022724  022740                          MCT33
7513  022726  104022                          ERROR   22                      ;TIME OUT ERROR...
7514  022730  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
7515  022734  000177  156502                  JMP     @NEXT                   ;GO TO NEXT TEST...
7516  022740                          MCT33:
7517  022740                          1$:     MOVE    # 0,BREG                ;SET TO CLEAR NPR IBUS+REGISTERS.
7518          000000                          MICPC=MICPC+1
7519  022740  000400                          .WORD   .$.
7520  022742                                  MOVE    BREG,SPAD <16>          ;FO RETURN ADDRESS PURPOSE...
7521          000001                          MICPC=MICPC+1
7522  022742  063236                          .WORD   .$.
7523  022744                                  MOVE    BREG,OUT0 <0>           ;CLEAR IBUS+IN DATA+LB.
7524          000002                          MICPC=MICPC+1
7525  022744  062220                          .WORD   .$.
7526  022746                                  MOVE    BREG,OUT0 <1>           ;CLEAR IBUS+IN DATA+HB.
7527          000003                          MICPC=MICPC+1
7528  022746  062221                          .WORD   .$.
7529  022750                                  MOVE    BREG,OUT0 <2>           ;CLEAR IBUS+OUT DATA+LB.
7530          000004                          MICPC=MICPC+1
7531  022750  062222                          .WORD   .$.
7532  022752                                  MOVE    BREG,OUT0 <3>           ;CLEAR IBUS+OUT DATA+HB.
7533          000005                          MICPC=MICPC+1
7534  022752  062223                          .WORD   .$.
7535  022754                                  MOVE    BREG,OUT0 <4>           ;CLEAR IBUS+IN BA+LB.
7536          000006                          MICPC=MICPC+1
7537  022754  062224                          .WORD   .$.
7538  022756                                  MOVE    BREG,OUT0 <5>           ;CLEAR IBUS+IN BA+HB.
7539          000007                          MICPC=MICPC+1
7540  022756  062225                          .WORD   .$.
7541  022760                                  MOVE    BREG,OUT0 <6>           ;CLEAR IBUS+OUT BA+LB.
7542          000010                          MICPC=MICPC+1
7543  022760  062226                          .WORD   .$.
7544  022762                                  MOVE    BREG,OUT0 <7>           ;CLEAR IBUS+OUT BA+HB.
7545          000011                          MICPC=MICPC+1
7546  022762  062227                          .WORD   .$.
7547  022764                                  MOVE    BREG,OUT1 <CSR10>       ;CLEAR IN+BA EABITS.
7548          000012                          MICPC=MICPC+1
7549  022764  061230                          .WORD   .$.
7550  022766                                  MOVE    INP1 <CSR11>,SPAD <0>   ;CLEAR OUT+BA EA+BITS.
7551          000013                          MICPC=MICPC+1
7552  022766  123220                          .WORD   .$.
7553  022770                                  OR      SPAD <0>,BREG,OUT1 <CSR11>      ;CLEAR OUT+BA EA+BITS.
7554          000014                          MICPC=MICPC+1
7555  022770  061311                          .WORD   .$.
7556  022772                                  MOVE    # 125,BREG              ; GOOD DATA.
7557          000015                          MICPC=MICPC+1
7558  022772  000525                          .WORD   .$.
7559  022774                                  MOVE    BREG,SPAD <3>           ;
7560          000016                          MICPC=MICPC+1
7561  022774  063223                          .WORD   .$.
7562  022776                                  SHFBRT                          ;
7563          000017                          MICPC=MICPC+1
```

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

```
7620
7621                    ;;      NOW GET IT BACK BY DATI.
7622 023044             ;;
7623         000042             MOVE    SPAD (1),BREG           ;SET IN+BA ADDRESS.
7624 023044 060601              MICPC=MICPC+1
7625 023046                     .WORD   .$.
7626         000043             MOVE    BREG,OUT0 (4)           ; "  "  "    "   .
7627 023046 062224              MICPC=MICPC+1
7630 023050                     .WORD   .$.
7631         000044             MOVE    SPAD (2),BREG           ; "  "  "    "   .
7632 023050 060602              MICPC=MICPC+1
7633 023052                     .WORD   .$.
7634         000045             MOVE    BREG,OUT0 (5)           ; "  "  "    "   .
7635 023052 062225              MICPC=MICPC+1
7636 023054                     .WORD   .$.
7637         000046             MOVE    # 001,BREG              ;SET NPR+RQ.
7640 023054 000401              MICPC=MICPC+1
7641 023056                     .WORD   .$.
7642         000047             MOVE    BREG,OUT1 (CSR10)       ;SET NPR+IN & NPR+RQ & WRD+XFR.
7643 023056 061230              MICPC=MICPC+1
7644 023060                     .WORD   .$.
7645         000050     45:     MOVE    INP1 (CSR10),BREG       ;IS NPR DONE??
7646 023060 120600              MICPC=MICPC+1
7647 023062                     .WORD   .$.
7650         000051             BBO     45                      ;NO, SPIN ON IT...
7651 023062 102050              MICPC=MICPC+1
7652 023064                     .WORD   .$.
7653         000052             MOVE    INP0 (0),BREG           ;GET IN DATA+LB.
7654 023064 020400              MICPC=MICPC+1
7655 023066                     .WORD   .$.
7656         000053             MOVE    INP0 (2),SPAD (0)       ;BRET OUT+DATA+LB.
7657 023066 023040              MICPC=MICPC+1
7660 023070                     .WORD   .$.
7661                            SIFEQ   BREG,SPAD (0) 6$        ;BRANCH IF GOOD..
7662
7663 023070
7664         000054             SUB2C   SPAD (0),BREG,NOP
7665 023070 060360              MICPC=MICPC+1
7666 023072                     .WORD   .$.
7667         000055             BZ      6$
7670 023072 101466              MICPC=MICPC+1
7671 023074                     .WORD   .$.
7672         000056             MOVE    BREG,OUT1 (CSR4)        ;GOOD DATA
7673 023074 061224              MICPC=MICPC+1
7674 023076                     .WORD   .$.
7675         000057             MOVE    SPAD (0),BREG           ;BAD DATA.
7676 023076 060600              MICPC=MICPC+1
7677 023100                     .WORD   .$.
7700         000060             MOVE    BREG,OUT1 (CSR5)        ;
7701 023100 061225              MICPC=MICPC+1
7702 023102                     .WORD   .$.
7703         000061             MOVE    # 11,MEM                ;SET ERROR TYPE...
7704 023102 002411              MICPC=MICPC+1
7705 023104                     .WORD   .$.
7706         000062             MOVE    MEM,OUT1 (CSR3)
7707 023104 041223              MICPC=MICPC+1
                                .WORD   .$.
```

```
023106                          CALL    ERORI
023106                          MOVE    # <MICPC+3>,BREG    ;DATA ERROR...
000063                          MICPC=MICPC+1
023106  000465                  .WORD   .$.
023110                          SBR     ERORI
000064                          MICPC=MICPC+1
023110  104401                  .WORD   .$.
023112                          SBR     1$                  ;LOOP ON ERROR...
000065                          MICPC=MICPC+1
023112  100400                  .WORD   .$.
023114                  6$:     MOVE    INPO <1>,BREG       ;GET IN DATA + HB....
000066                          MICPC=MICPC+1
023114  020420                  .WORD   .$.
023116                          MOVE    INPO <3>,SPAD <0>   ;GET OUT+DATA+HB....
000067                          MICPC=MICPC+1
023116  023060                  .WORD   .$.
023120                          $IFEQ   BREG,SPAD <0> 7$    ;BRANCH IF GOOD...


023120                          SUB2C   SPAD <0>,BREG,NOP
000070                          MICPC=MICPC+1
023120  060360                  .WORD   .$.
023122                          BZ      7$
000071                          MICPC=MICPC+1
023122  101502                  .WORD   .$.
023124                          MOVE    BREG,OUT1 <CSR4>    ;GOOD DATA.
000072                          MICPC=MICPC+1
023124  061224                  .WORD   .$.
023126                          MOVE    SPAD <0>,BREG       ;BAD DATA.
000073                          MICPC=MICPC+1
023126  060600                  .WORD   .$.
023130                          MOVE    BREG,OUT1 <CSR5>    ;
000074                          MICPC=MICPC+1
023130  061225                  .WORD   .$.
023132                          MOVE    # 11,MEM            ;SET ERROR TYPE...
000075                          MICPC=MICPC+1
023132  002411                  .WORD   .$.
023134                          MOVE    MEM,OUT1 <CSR3>     ;
000076                          MICPC=MICPC+1
023134  041223                  .WORD   .$.
023136                          CALL    ERORI               ;DATA ERROR...
023136                          MOVE    # <MICPC+3>,BREG
000077                          MICPC=MICPC+1
023136  000501                  .WORD   .$.
023140                          SBR     ERORI
000100                          MICPC=MICPC+1
023140  104401                  .WORD   .$.
023142                          SBR     1$                  ;LOOP ON ERROR...
000101                          MICPC=MICPC+1
023142  100400                  .WORD   .$.
023144                  7$:     CALL    SCPE                ;SCOPE THE TEST...
023144                          MOVE    # <MICPC+3>,BREG
000102                          MICPC=MICPC+1
023144  000504                  .WORD   .$.
023146                          SBR     SCPE
000103                          MICPC=MICPC+1
```

```
023146  104454                  .WORD   .$.
023150                          SBR     1$              ;YES, DO NEXT ITERATION.
        000104                  MICPC=MICPC+1
023150  100400                  .WORD   .$.
023152  000000          NPRDTI: .WORD   0               ;
023154                  SNPR9   21
023154                  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
023154                  SXZ
                        ;:*************************************************************
023154                  STSTN
                        ;  TEST 34
                        ;-------------
023154  012737  000034  001202  TST34:  MOV     #34,$TSTNM              ; LOAD THE NO. OF THIS TEST
023162  012737  023322  001442          MOV     #TST35,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
023170  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
023174  023210                          MCT34
023176  104022                          ERROR   22                      ;TIME OUT ERROR...
023200  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
023204  000177  156232                  JMP     @NEXT                   ;GO TO NEXT TEST...
023210                          MCT34:
023210                          1$:     MOVE    # 0,BREG               ;SET TO CLEAR IBUS+REGISTERS.
        000000                  MICPC=MICPC+1
023210  000400                  .WORD   .$.
023212                          MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES.
        000001                  MICPC=MICPC+1
023212  063236                  .WORD   .$.
023214                          MOVE    BREG,SPAD <1>   ;CLEAR DELAY COUNT...
        000002                  MICPC=MICPC+1
023214  063221                  .WORD   .$.
023216                          MOVE    BREG,OUTD <0>
        000003                  MICPC=MICPC+1
023216  062220                  .WORD   .$.
023220                          MOVE    BREG,OUTD <1>   ;
        000004                  MICPC=MICPC+1
023220  062221                  .WORD   .$.
023222                          MOVE    BREG,OUTD <2>   ;
        000005                  MICPC=MICPC+1
023222  062222                  .WORD   .$.
023224                          MOVE    BREG,OUTD <3>   ;
        000006                  MICPC=MICPC+1
023224  062223                  .WORD   .$.
023226                          MOVE    BREG,OUTD <4>   ;
        000007                  MICPC=MICPC+1
023226  062224                  .WORD   .$.
023230                          MOVE    BREG,OUTD <5>   ;
        000010                  MICPC=MICPC+1
023230  062225                  .WORD   .$.
023232                          MOVE    BREG,OUTD <6>   ;
```

```
                000011                          MICPC=MICPC+1
7789  023232    062226                          .WORD   .$.
7790  023234                                    MOVE    BREG,OUT0 <7>   ;
                000012                          MICPC=MICPC+1
7792  023234    062227                          .WORD   .$.
7793  023236                                    MOVE    BREG,OUT1 <CSR10>       ;
                000013                          MICPC=MICPC+1
7795  023236    061230                          .WORD   .$.
7796  023240                          2$:       MOVE    # 320,BREG      ;NPR ADDRESS+LB.
                000014                          MICPC=MICPC+1
7798  023240    000170                          .WORD   .$.
7799  023242                          ;E        BREG,OUT0 <6>   ;LOAD OUT+BA+LB.
7800  023242    063244                          MICPC=MICPC+1
7801  023244                                    .WORD   .$.
                000016                          MOVE    # 376,BREG      ;NPR ADDRESS+HB.
7803  023244    000776                          MICPC=MICPC+1
7804  023246                                    .WORD   .$.
                000017                  ;E      BREG,OUT0 <7>   ;LOAD OUT+BA+HB.
7806  023246    062227                          MICPC=MICPC+1
7807  023250                                    .WORD   .$.
                000020                          MOVE    # 14,MEM                ;SET OUT+BA BITS 16 &17.
7809  023250    002414                          MICPC=MICPC+1
7810  023252                                    .WORD   .$.
                000021                          MOVE    INP1 <CSR11>,SPAD <0>   ;SET OUT+BA 16+17
7812  023252    123220                          MICPC=MICPC+1
7813  023254                                    .WORD   .$.
                000022                          OR      MEM,SPAD <0>,OUT1 <CSR11>       ;SET OUT+BA 16 &17.
7815  023254    041311                          MICPC=MICPC+1
7816  023256                                    .WORD   .$.
                000023                          MOVE    # 21,BREG       ;
7818  023256    000421                          MICPC=MICPC+1
7819  023260                                    .WORD   .$.
                000024                          MOVE    BREG,OUT1 <CSR10>       ;SET NPR+RQ OUT+NPR,WRD+XFR...
7821  023260    061230                          MICPC=MICPC+1
7822  023262                                    .WORD   .$.
                000025                  3$:     MOVE    INP1 <CSR11>,BREG       ;CHECK IF NON+EX+MEM SET...
7824  023262    120620                          MICPC=MICPC+1
7825  023264                                    .WORD   .$.
                000026                          BBO     4$              ;YES, ITS WORKING.
7827  023264    102037                          MICPC=MICPC+1
7828  023266                                    .WORD   .$.
                000027                          SINC    SPAD <1>        ;WAIT FOR ATLEAST 1BUSEC.
7830  023266    063061                          MICPC=MICPC+1
7831  023270                          .WORD     .$.!.DSP
                000030                          BC      6$              ;ENOUGH REPORT ERROR...
7833  023270    101032                          MICPC=MICPC+1
7834  023272                                    .WORD   .$.
                000031                          SBR     3$              ;CHECK AGAIN...
7836  023272    100425                          MICPC=MICPC+1
7837  023274                                    .WORD   .$.
                000032                  6$:     MOVE    # 13,MEM                ;SET ERROR TYPE.
7839  023274    002413                          MICPC=MICPC+1
7840  023276                                    .WORD   .$.
                000033                          MOVE    MEM,OUT1 <CSR3>         ;
7842  023276    041223                          MICPC=MICPC+1
7843                                            .WORD   .$.
```

```
7844  023300                                    CALL    ERORI           ;REPORT ERROR...
7845  023300                                    MOVE    # <MICPC+3>,BREG
7846                 000034                    MICPC=MICPC+1
7847  023300 000436                            .WORD   .$.
7848  023302                                    SBR     ERORI
7849                 000035                    MICPC=MICPC+1
7850  023302 104401                            .WORD   .$.
7851  023304                                    SBR     1$              ;LOOP ON ERROR.
7852                 000036                    MICPC=MICPC+1
7853  023304 100400                            .WORD   .$.
7854  023306                             4$:    MOVE    # 0,BREG                ;
7855                 000037                    MICPC=MICPC+1
7856  023306 000400                            .WORD   .$.
7857  023310                                    MOVE    INP1 <CSR11>,SPAD <0>   ;
7858                 000040                    MICPC=MICPC+1
7859  023310 123220                            .WORD   .$.
7860  023312                                    AND     SPAD <0>,BREG,OUT1 <CSR11>      ;RESET NON-EX-MEM.
7861                 000041                    MICPC=MICPC+1
7862  023312 061271                            .WORD   .$.
7863  023314                                    CALL    SCPE            ;SCOPE THE TEST.
7864  023314                                    MOVE    # <MICPC+3>,BREG
7865                 000042                    MICPC=MICPC+1
7866  023314 000444                            .WORD   .$.
7867  023316                                    SBR     SCPE
7868                 000043                    MICPC=MICPC+1
7869  023316 104454                            .WORD   .$.
7870  023320                                    SBR     1$              ;DO NEXT ITERATION.
7871                 000044                    MICPC=MICPC+1
7872  023320 100400                            .WORD   .$.
7873  023322                             SNPR4
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                             $TSTN
7886                                           ;  TEST 35
7887                                           ;-------------
7888  023322 012737 000035 001202       TST35: MOV     #35,$TSTNM              ; 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,LDVRWT               ;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  023356                             MCT35:
7897  023356                             1$:    MOVE    # 5,SPAD <5>     ;SET THE COUNT
7898                 000000                    MICPC=MICPC+1
7899  023356 003005                            .WORD   .$.
```

```
7900 023360                     MOVE    # 13,SPAD <13>  ;
7901         000001             MICPC=MICPC+1
7902 023360 003013              .WORD   .$.
7903 023362                     MOVE    # 0,BREG        ;SET TO CLEAR NPR IBNS+REGISTERS.
7904         000002             MICPC=MICPC+1
7905 023362 000400              .WORD   .$.
7906 023364                     MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
7907         000003             MICPC=MICPC+1
7908 023364 063236              .WORD   .$.
7909 023366                     MOVE    BREG,OUTO <0>   ;CLEAR IBUS
7910         000004             MICPC=MICPC+1
7911 023366 062220              .WORD   .$.
7912 023370                     MOVE    BREG,OUTO <1>   ;CLEAR IBUS.
7913         000005             MICPC=MICPC+1
7914 023370 062221              .WORD   .$.
7915 023372                     MOVE    BREG,OUTO <4>   ;CLEAR IBUS.
7916         000006             MICPC=MICPC+1
7917 023372 062224              .WORD   .$.
7918 023374                     MOVE    BREG,OUTO <5>   ;CLEAR IBUS.
7919         000007             MICPC=MICPC+1
7920 023374 062225              .WORD   .$.
7921 023376                     MOVE    BREG,OUT1 <CSR10>       ;CLEAR EA+BITS
7922         000010             MICPC=MICPC+1
7923 023376 061230              .WORD   .$.
7924 023400                     MOVE    # <MNPDT&377>,BREG      ;SET NPR ADDRESS.
7925         000011             MICPC=MICPC+1
7926 023400 000572              .WORD   .$.
7927 023402                     MOVE    BREG,SPAD <1>   ;
7928         000012             MICPC=MICPC+1
7929 023402 063221              .WORD   .$.
7930 023404                     MOVE    # <MNPDT&177400/400>,BREG       ;
7931         000013             MICPC=MICPC+1
7932 023404 000447              .WORD   .$.
7933 023406                     MOVE    BREG,SPAD <2>           ;
7934         000014             MICPC=MICPC+1
7935 023406 063222              .WORD   .$.
7936 023410                     MOVE    # 125,BREG              ;GET READY TO LOAD MEMORY.
7937         000015             MICPC=MICPC+1
7938 023410 000525              .WORD   .$.
7939 023412                     MOVE    # 0,MLR                 ;
7940         000016             MICPC=MICPC+1
7941 023412 010000              .WORD   .$.
7942 023414                     MOVE    # 2,MPR          ;
7943         000017             MICPC=MICPC+1
7944 023414 004002              .WORD   .$.
7945 023416             125:    MOVE    BREG,MEM,MARINC         ;LOAD THE MEMORY...
7946         000020             MICPC=MICPC+1
7947 023416 076620              .WORD   .$.
7948 023420                     SHFBRT                          ;SET ALTERNATE 125 OR 252.
7949         000021             MICPC=MICPC+1
7950 023420 061620              .WORD   .SBR!.SELB!.DBRSH
7951 023422                     SDEC    SPAD <13>               ;IS IT DONE???
7952         000022             MICPC=MICPC+1
7953 023422 063173      .WORD   .$.!.DSP
7954 023424                     BZ      15$                     ;YES, START THE TEST.
7955         000023             MICPC=MICPC+1
```

```
023424 101425                   .WORD   .$.
023426                          SBR     12$                     ;NO, CONTINUE.
000024                          MICPC=MICPC+1
023426 100420                   .WORD   .$.
023430                  15$:    MOVE    # 0,MLR         ;
000025                          MICPC=MICPC+1
023430 010000                   .WORD   .$.
023432                          MOVE    # 0,MPR         ;
000026                          MICPC=MICPC+1
023432 004000                   .WORD   .$.
023434                  2$:     MOVE    SPAD (1),BREG   ;SET IN BA ADDRESS.
000027                          MICPC=MICPC+1
023434 060601                   .WORD   .$.
023436                          MOVE    BREG,OUTO (4)   ;SET IN+BA ADDRESS.
000030                          MICPC=MICPC+1
023436 062224                   .WORD   .$.
023440                          MOVE    SPAD (2),BREG   ;SET IN+BA ADDRESS.
000031                          MICPC=MICPC+1
023440 060602                   .WORD   .$.
023442                          MOVE    BREG,OUTO (5)   ;SET IN+BA ADDRESS.
000032                          MICPC=MICPC+1
023442 062225                   .WORD   .$.
023444                          MOVE    # 03,BREG       ;SET NOT+LAST+XFR & NPR.R1
000033                          MICPC=MICPC+1
023444 000403                   .WORD   .$.
023446                          MOVE    BREG,OUT1 (CSR10)       ;SET NOT+LAST+XFR & NPR R1 & WRB.XFR.
000034                          MICPC=MICPC+1
023446 061230                   .WORD   .$.
023450                  3$:     MOVE    INP1 (CSR10),BREG       ;IS NPR DONE??
000035                          MICPC=MICPC+1
023450 120600                   .WORD   .$.
023452                          BBO     3$              ;NO,SPIN ON IT
000036                          MICPC=MICPC+1
023452 102035                   .WORD   .$.
023454                          MOVE    INPO (0),MEM MARINC     ;
000037                          MICPC=MICPC+1
023454 036400                   .WORD   .$.
023456                          MOVE    INPO (1),MEM MARINC     ;
000040                          MICPC=MICPC+1
023456 036420                   .WORD   .$.
023460                          SDEC    SPAD (5)        ;IS IT DONE??
000041                          MICPC=MICPC+1
023460 063165           .WORD   .$.!.DSP
023462                          BZ      4$              ;YES, GO CHECK THE DATA.
000042                          MICPC=MICPC+1
023462 101447                   .WORD   .$.
023464                          SINC    SPAD (1)                ;UPDATE ADDRESS.
000043                          MICPC=MICPC+1
023464 063061           .WORD   .$.!.DSP
023466                          SINC    SPAD (1)                ;
000044                          MICPC=MICPC+1
023466 063061           .WORD   .$.!.DSP
023470                          SADC    SPAD (2)                ;UPDATE ADDRESS.
000045                          MICPC=MICPC+1
023470 063102           .WORD   .$.!.DSP
023472                          SBR     2$              ;DO NEXT NPR.
```

```
8012                    000046
8013  023472  100427            MICPC=MICPC+1
8014  023474                    .WORD   .$.
                        45:     MOVE    # 0,BREG                 ;MOVE # 5,SPAD (8)
8015                    000047  MICPC=MICPC+1
8016  023474  000400            .WORD   .$.
8017  023476                    MOVE    # 13,SPAD (13)           ;
8018                    000050  MICPC=MICPC+1
8019  023476  003013            .WORD   .$.
8020  023600                    MOVE    BREG,OUT1 (CSR10)        ;CLEAR NOT-LAST-XFR & NPR RQ.
8021                    000051  MICPC=MICPC+1
8022  023600  061230            .WORD   .$.
8023  023602                    MOVE    BREG,SPAD (3)    ;
8024                    000052  MICPC=MICPC+1
8025  023602  063223            .WORD   .$.
8026  023604                    MOVE    BREG,SPAD (0)
8027                    000053  MICPC=MICPC+1
8028  023604  063220            .WORD   .$.
8029  023606                    MOVE    BREG,SPAD (1)
8030                    000054  MICPC=MICPC+1
8031  023606  063221            .WORD   .$.
8032  023610                    MOVE    # 2,SPAD (2)     ;
8033                    000055  MICPC=MICPC+1
8034  023610  003002            .WORD   .$.
8035  023612                    MOVE    # 0,MLR          ;RESET MAR.
8036                    000056  MICPC=MICPC+1
8037  023612  010000            .WORD   .$.
8038  023614                    MOVE    # 0,MPR          ;RESTE MAR.
8039                    000057  MICPC=MICPC+1
8040  023614  004000            .WORD   .$.
8041  023616            55:     MOVE    MEM,SPAD (0)     ;GET THE NPR DATA.
8042                    000060  MICPC=MICPC+1
8043  023616  043220            .WORD   .$.
8044  023620                    MOVE    SPAD (1),MLR     ;SET GOOD DATA ADDRESS.
8045                    000061  MICPC=MICPC+1
8046  023620  070201            .WORD   .$.
8047  023622                    MOVE    SPAD (2),MPR     ;SET GOOD DATA ADDRESS.
8048                    000062  MICPC=MICPC+1
8049  023622  064202            .WORD   .$.
8050  023624                    SIFEQ   MEM,SPAD (0) 65$ ;IF GOOD CHECK NEXT...
8051
8052
8053  023624                    SUB2C   SPAD (0),MEM,NOP
8054                    000063  MICPC=MICPC+1
8055  023624  040360            .WORD   .$.
8056  023626                    BZ      65$
8057                    000064  MICPC=MICPC+1
8058  023626  101475            .WORD   .$.
8059  023630                    MOVE    MEM,OUT1 (CSR4)          ;GOOD DATA.
8060                    000065  MICPC=MICPC+1
8061  023630  041224            .WORD   .$.
8062  023632                    MOVE    SPAD (0),BREG    ;BAD DATA.
8063                    000066  MICPC=MICPC+1
8064  023632  060600            .WORD   .$.
8065  023634                    MOVE    BREG,OUT1 (CSR5)         ;
8066                    000067  MICPC=MICPC+1
8067  023634  061225            .WORD   .$.
```

```
023536                          MOVE    # 11,MEM        ;ERROR TYPE...
000070                          MICPC=MICPC+1
023536 002411                   .WORD   .$.
023540                          MOVE    MEM,OUT1 <CSR3>;
000071                          MICPC=MICPC+1
023540 041223                   .WORD   .$.
023542                          CALL    EROR1           ;DATA ERROR...
023542                          MOVE    # <MICPC+3>,BREG
000072                          MICPC=MICPC+1
023542 000474                   .WORD   .$.
023544                          SBR     ERROR1
000073                          MICPC=MICPC+1
023544 104401                   .WORD   .$.
023546                          SBR     1$              ;LOOP ON ERROR...
000074                          MICPC=MICPC+1
023546 100400                   .WORD   .$.
023550                  6$:     $DEC    SPAD <13>               ;IS IT DONE??
000075                          MICPC=MICPC+1
023550 063173           .WORD   .$.!.DSP
023552                          BZ      7$              ;YES, SCOPE IT.
000076                          MICPC=MICPC+1
023552 101503                   .WORD   .$.
023554                          $INC    SPAD <3>,,MLRLD ;NPR DATA ADDRESS.
000077                          MICPC=MICPC+1
023554 073063           .WORD   .$.!.DSP!MLRLD
023556                          $INC    SPAD <1>                ;UPDATE GOOD DATA ADDRESS.
000100                          MICPC=MICPC+1
023556 063061           .WORD   .$.!.DSP
023560                          $ADC    SPAD <2>                ;UPDATA GOOD DATA ADDRESS.
000101                          MICPC=MICPC+1
023560 063102           .WORD   .$.!.DSP
023562                          SBR     5$              ;CHECK NEXT BYTE.
000102                          MICPC=MICPC+1
023562 100460                   .WORD   .$.
023564                  7$:     CALL    SCPE            ;SCOPE IT.
023564                          MOVE    # <MICPC+3>,BREG
000103                          MICPC=MICPC+1
023564 000505                   .WORD   .$.
023566                          SBR     SCPE
000104                          MICPC=MICPC+1
023566 104454                   .WORD   .$.
023570                          SBR     1$              ;DO NEXT ITERATION.
000105                          MICPC=MICPC+1
023570 100400                   .WORD   .$.
023572 125125 125125 125125 MNPDT:  .WORD   125125,125125,125125        ;
023600 125125 125125 125125         .WORD   125125,125125,125125        ;
                        ;MNPDT1:            .BLKW   5
023606                  $MEM10
023606                  $XZ


                        ;************************** TEST 36 **************************
                        ;* MAIN MEMORY TEST
                        ;* FLOAT A 0 THROUGH ALL MAIN MEMORY LOCATION...
023606                  $XZ
                        ;;*************************************************************
```

```
8124
8125  023606                                  $TSTN
8126                                                  ;  TEST 36
8127                                                  ;-------------
8128  023606  012737  000036  001202  TST36:  MOV     #36,$TSTNM              ; 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,LDVRWT               ;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                          $MMFLT  0,11$,12$,13$,14$,15$
8138  023642                                  MOVE    # 0,BREG                ;SET TO CLEAR SPAD 16...
8139          000000                          MICPC=MICPC+1
8140  023642  000400                          .WORD   .$.
8141  023644                                  MOVE    BREG,SPAD (16)          ;FOR RETURN ADDRESS PURPOSES...
8142          000001                          MICPC=MICPC+1
8143  023644  063236                          .WORD   .$.
8144  023646                          11$:    MOVE    # 177,BREG              ;START WITH BIT 7.
8145          000002                          MICPC=MICPC+1
8146  023646  000577                          .WORD   .$.
8147  023650                                  MOVE    # 0,MLR                 ;LOAD MAR+LO.
8148          000003                          MICPC=MICPC+1
8149  023650  010000                          .WORD   .$.
8150  023652                                  MOVE    # 0,MPR                 ;LOAD MAR+HI.
8151          000004                          MICPC=MICPC+1
8152  023652  004000                          .WORD   .$.
8153  023654                                  MOVE    # 0,MEM
8154          000005                          MICPC=MICPC+1
8155  023654  002400                          .WORD   .$.
8156  023656                                  MOVE    MEM,SPAD (1)            ;SET THE COUNTER...
8157          000006                          MICPC=MICPC+1
8158  023656  043221                          .WORD   .$.
8159  023660                                  MOVE    MEM,SPAD (3)            ;SET THE COUNTER...
8160          000007                          MICPC=MICPC+1
8161  023660  043223                          .WORD   .$.
8162
8163  023662                          12$:    MOVE    BREG,MEM                ;SET THE BIT IN MEMORY.
8164          000010                          MICPC=MICPC+1
8165  023662  062620                          .WORD   .$.
8166  023664                                  MOVE    MEM,SPAD (2)            ;PUT THE "FOUND" IN SPAD  (2).
8167          000011                          MICPC=MICPC+1
8168  023664  043222                          .WORD   .$.
8169  023666                                  MOVE    BREG,SPAD (4)           ;SAVE THE BREG.
8170          000012                          MICPC=MICPC+1
8171  023666  063224                          .WORD   .$.
8172  023670                                  $IFEQ   BREG,SPAD (2)   13$     ;IF GOOD CONTINUE.
8173
8174
8175  023670                                  SUB2C   SPAD (2),BREG,NOP
8176          000013                          MICPC=MICPC+1
8177  023670  060362                          .WORD   .$.
8178  023672                                  BZ      13$
8179          000014                          MICPC=MICPC+1
```

```
8180  023672  101427
8181  023674
8182          000015
8183  023674  061224
8184  023676
8185          000016
8186  023676  041225
8187  023700
8188          000017
8189  023700  000406
8190  023702
8191          000020
8192  023702  061223
8193  023704
8194          000021
8195  023704  060601
8196  023706
8197          000022
8198  023706  061227
8199  023710
8200  023710
8201          000023
8202  023710  000425
8203  023712
8204          000024
8205  023712  104400
8206  023714
8207          000025
8208  023714  060604
8209  023716
8210          000026
8211  023716  100410
8212  023720
8213  023720
8214          000027
8215  023720  000431
8216  023722
8217          000030
8218  023722  104427
8219  023724
8220          000031
8221  023724  060604
8222  023726
8223          000032
8224  023726  100410
8225  023730
8226          000033
8227  023730  060604
8228  023732
8229          000034
8230  023732  061620
8231  023734
8232          000035
8233  023734  103410
8234  023736
8235          000036
```

```
        .WORD   .$.
        MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,OUT1 <CSR5> ;BAD DATA...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 6,BREG                ;SET TYPE OF ERROR...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 <CSR3>        ;
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD <1>,BREG           ;LOAD ADDRESS.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 <CSR7>        ;
        MICPC=MICPC+1
        .WORD   .$.
        CALL    ERROR                   ;REPORT ERROR...
        MOVE    # <MICPC+3>,BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     ERROR
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD <4>,BREG           ;RESTORE BREGISTER...
        MICPC=MICPC+1
        .WORD   .$.
        SBR     12$                     ;LOOP ON ERROR...
        MICPC=MICPC+1
        .WORD   .$.
13$:    CALL    SCP1                    ;LOOP ON DATA SET??
        MOVE    # <MICPC+3>,BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     SCP1
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD <4>,BREG           ;YES, FIRST RESTORE BREG...
        MICPC=MICPC+1
        .WORD   .$.
        SBR     12$                     ;SCOPE THE DATA..
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD <4>,BREG           ;RESTORE BREG...
        MICPC=MICPC+1
        .WORD   .$.
        SHFRT                           ;RIGHT SHIFT BY 1 BIT.
        MICPC=MICPC+1
        .WORD   .SBR!.SELB!.DBRSH
B87     12$             ;BRANCH IF NOT DONE.
        MICPC=MICPC+1
        .WORD   .$.
14$:    $INC    SPAD <1>                ;INCREMENT ADDRESS.
        MICPC=MICPC+1
```

```
8236  023736  063061          .WORD   .$.!.DSP
8237  023740                          $AOC    SPAD (3)                        ;  "      "       .
                000037                  MICPC=MICPC+1
8239  023740  063103          .WORD   .$.!.DSP
8240  023742                          MOVE    # 4,MEM                 ;PREPARE TO CHECK DONE...
                000040                  MICPC=MICPC+1
8242  023742  002404                  .WORD   .$.
8243  023744                          $IFLO   MEM,SPAD (3)    15$     ;IS IT DONE???
8244
8245
8246  023744                          SUB2C   SPAD (3),MEM,NOP
8247            000041                MICPC=MICPC+1
8248  023744  040363                  .WORD   .$.
8249  023746                          BC      15$
8250            000042                MICPC=MICPC+1
8251  023746  101046                  .WORD   .$.
8252
8253  023750                          MOVE    SPAD (1),MLR            ;LOAD MAR+LO.
8254            000043                MICPC=MICPC+1
8255  023750  070201                  .WORD   .$.
8256  023752                          MOVE    SPAD (3),MPR            ;LOAD MAR+HI.
8257            000044                MICPC=MICPC+1
8258  023752  064203                  .WORD   .$.
8259  023754                          SBR     12$                     ;DO THE NEXT MEMORY LOCATION.
8260            000045                MICPC=MICPC+1
8261  023754  100410                  .WORD   .$.
8262  023756                  15$:    CALL    SCPE                    ;SCOPE THE TEST...
8263  023756                          MOVE    # (MICPC+3),BREG
8264            000046                MICPC=MICPC+1
8265  023756  000450                  .WORD   .$.
8266  023760                          SBR     SCPE
8267            000047                MICPC=MICPC+1
8268  023760  104454                  .WORD   .$.
8269  023762                          SBR     11$
8270            000050                MICPC=MICPC+1
8271  023762  100402                  .WORD   .$.
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                  $TSTN
8283                                    ;  TEST 37
8284                                    ;-------------
8285  023764  012737  000037  001202  TST37:  MOV     #37,$TSTNM              ; LOAD THE NO. OF THIS TEST
8286  023772  012737  024144  001442          MOV     #TST40,NEXT             ; POINT TO THE START OF NEXT TEST.
8287                                                                  ;R1 CONTAINS BASE 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...
```

```
                                        JMP     @NEXT                   ;GO TO NEXT TEST...
                                MCT37:
                                SWAFLT  1,1$,2$,3$,4$,5$
                                        MOVE    # 0,BREG                ;SET TO CLEAR SPAD 16...
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS PURPOSES...
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                1$: MOVE        # 200,BREG              ;START WITH BIT 7.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    # 0,MLR                  ;LOAD MAR<-LO.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    # 0,MPR                  ;LOAD MAR<-HI.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    # 0,MEM
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    MEM,SPAD <1>            ;SET THE COUNTER...
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    MEM,SPAD <3>            ;SET THE COUNTER...
                                        MICPC=MICPC+1
                                        .WORD   .$.

                                2$:     MOVE    BREG,MEM                ;SET THE BIT IN MEMORY.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    MEM,SPAD <2>            ;PUT THE "FOUND" IN SPAD  <2>.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    BREG,SPAD <4>           ;SAVE THE BREG.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        $IFEQ   BREG,SPAD <2>   3$      ;IF GOOD CONTINUE.


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

```
                        000020
024060  061223
024062
                        000021
024062  060601
024064
                        000022
024064  061227
024066
024066
                        000023
024066  000425
024070
                        000024
024070  104400
024072
                        000025
024072  060604
024074
                        000026
024074  100410
024076
024076
                        000027
024076  000431
024100
                        000030
024100  104427
024102
                        000031
024102  060604
024104
                        000032
024104  100410
024106
                        000033
024106  060604
024110
                        000034
024110  061620
024112
                        000035
024112  103437
024114
                        000036
024114  100410
024116
                        000037
024116  063061
024120
                        000040
024120  063103
024122
                        000041
024122  002404
024124
```

```
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD (1),BREG           ;LOAD ADDRESS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT1 (CSR7)        ;
                MICPC=MICPC+1
                .WORD   .$.
                CALL    EROR                    ;REPORT ERROR...
                MOVE    # (MICPC+3),BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     EROR
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD (4),BREG           ;RESTORE BREGISTER...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     2$                      ;LOOP ON ERROR...
                MICPC=MICPC+1
                .WORD   .$.
3$:             CALL    SCP1                    ;LOOP ON DATA SET??
                MOVE    # (MICPC+3),BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     SCP1
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD (4),BREG           ;YES, FIRST RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     2$                      ;SCOPE THE DATA..
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD (4),BREG           ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$.
                SHFRT                           ;RIGHT SHIFT BY 1 BIT.
                MICPC=MICPC+1
                .WORD   .SBR!.SELB!.DBRSH
                BR      4$                      ;BRANCH IF DONE...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     2$                      ;ELSE, CONTINUE.
                MICPC=MICPC+1
                .WORD   .$.
4$:             SINC    SPAD (1)                        ;INCREMENT ADDRESS.
                MICPC=MICPC+1
.WORD           .$.!.DSP
                SADC    SPAD (3)                        ;   "       "       .
                MICPC=MICPC+1
.WORD           .$.!.DSP
                MOVE    # 4,MEM                 ;PREPARE TO CHECK DONE...
                MICPC=MICPC+1
                .WORD   .$.
                SIFLO   MEM,SPAD (3)    5$      ;IS IT DONE???
```

```
                                        SUB2C   SPAD (3),MEM,NOP
                                        MICPC=MICPC+1
024124                                  .WORD   .$.
          000042
024124  040363                        BC      5$
024126                                MICPC=MICPC+1
          000043
024126  101047                        .WORD   .$.

024130                                MOVE    SPAD (1),MLR            ;LOAD MAR+LO.
          000044                      MICPC=MICPC+1
024130  070201                        .WORD   .$.
024132                                MOVE    SPAD (3),MPR            ;LOAD MAR+HI.
          000045                      MICPC=MICPC+1
024132  064203                        .WORD   .$.
024134                                SBR     2$                      ;DO THE NEXT MEMORY LOCATION.
          000046                      MICPC=MICPC+1
024134  100410                        .WORD   .$.
024136                          5$:   CALL    SCPE                    ;SCOPE THE TEST...
024136                                MOVE    @ (MICPC+3),BREG
          000047                      MIC `=MICPC+1
024136  000451                        .WORD   .$.
024140                                SBR     SCPE
          000050                      MICPC=MICPC+1
024140  104454                        .WORD   .$.
024142                                SBR     1$
          000051                      MICPC=MICPC+1
024142  100402                        .WORD   .$.
024144                          $MEM2
024144                          $XZ


                                        ;************************** 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.
024144                          $XZ
                                        ;;***********************************************************************

024144                          $TSTN
                                        ;  TEST 40
                                        ;--------------
024144  012737  000040  001202  TST40:  MOV     #40,$TSTNM              ; LOAD THE NO. OF THIS TEST
024152  012737  024362  001442          MOV     #TST41,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
024160  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
024164  024200                          MCT40
024166  104022                          ERROR   22                      ;TIME OUT ERROR...
024170  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
024174  000177  155242                  JMP     @NEXT                   ;GO TO NEXT TEST...
024200                          MCT40:
024200                          6$:     MOVE    # 0,BREG                ;START AT ADDRESS 0.
          000000                        MICPC=MICPC+1
024200  000400                          .WORD   .$.
024202                                  MOVE    BREG,SPAD (16)          ;FOR RETURN ADDRESS PURPOSE...
          000001                        MICPC=MICPC+1
```

```
                                .WORD   .$
                                MOVE    #0,MLR                  ;LOAD MAR<-LO.
                                MICPC=MICPC+1
                                .WORD   .$
                                MOVE    #0,MPR                  ;LOAD MAR<-HI.
                                MICPC=MICPC+1
                                .WORD   .$
                                MOVE    BREG,SPAD (1)           ;SET THE PARALLEL ADDRESS.
                                MICPC=MICPC+1
                                .WORD   .$
                                MOVE    BREG,SPAD (2)           ;SET THE PARALLEL ADDRESS.
                                MICPC=MICPC+1
                                .WORD   .$
                        1$:     MOVE    SPAD (1),MEM            ;SET THE ADDRESS IN MEMORY.
                                MICPC=MICPC+1
                                .WORD   .$
                                MOVE    MEM,BREG                ;PUT FOUND IN BREG.
                                MICPC=MICPC+1
                                .WORD   .$
                                SIFEQ   BREG,SPAD (1)   2$      ;BRANCH IF GOOD...


                                SUB2C   SPAD (1),BREG,NOP
                                MICPC=MICPC+1
                                .WORD   .$
                                BZ      2$
                                MICPC=MICPC+1
                                .WORD   .$
                                MOVE    MEM,OUT1 (CSR5)         ;LOAD GOOD DATA...
                                MICPC=MICPC+1
                                .WORD   .$
                                MOVE    SPAD (1),BREG           ;LOAD BAD DATA...
                                MICPC=MICPC+1
                                .WORD   .$
                                MOVE    BREG,OUT1 (CSR4)        ;
                                MICPC=MICPC+1
                                .WORD   .$
                                MOVE    #7,BREG                 ;SET THE ERROR TYPE...
                                MICPC=MICPC+1
                                .WORD   .$
                                MOVE    BREG,OUT1 (CSR3)        ;
                                MICPC=MICPC+1
                                .WORD   .$
                                MOVE    SPAD (1),BREG           ;LOAD ADDRESS.
                                MICPC=MICPC+1
                                .WORD   .$
                                MOVE    BREG,OUT1 (CSR7)        ;
                                MICPC=MICPC+1
                                .WORD   .$
                                CALL    EROR1                   ;REPORT ERROR...
                                MOVE    #(MICPC+3),BREG
                                MICPC=MICPC+1
                                .WORD   .$
                                $BR     EROR1
                                MICPC=MICPC+1
                                .WORD   .$
```

```
8516  024246                    SBR     1$                          ;LOOP ON ERROR...
8517          000023            MICPC=MICPC+1
8518  024246  100406            .WORD   .$.
8519  024250                2$: CALL    SCP11                       ;SCOPE ON DATA SET?
8520  024250                    MOVE    @ (MICPC+3),BREG
8521          000024            MICPC=MICPC+1
8522  024250  000426            .WORD   .$.
8523  024252                    SBR     SCP11
8524          000025            MICPC=MICPC+1
8525  024252  104430            .WORD   .$.
8526  024254                    SBR     1$                          ;YES, DO IT...
8527          000026            MICPC=MICPC+1
8528  024254  100406            .WORD   .$.
8529  024256                    $INC    SPAD (1),       ,MLRLD      ;INCREMENT THE ADDRESS.
8530          000027            MICPC=MICPC+1
8531  024256  073061      .WORD .$.!.DSP!MLRLD
8532  024260                    $ADC    SPAD (2),       ,MPRLD      ;INCREMENT PAGE ADDRESS.
8533          000030            MICPC=MICPC+1
8534  024260  067102      .WORD .$.!.DSP!MPRLD
8535  024262                    MOVE    @ 4,BREG                            ;CHECK IF DONE?
8536          000031            MICPC=MICPC+1
8537  024262  000404            .WORD   .$.
8538  024264                    $IFHI   BREG,SPAD (2)   1$          ;NO, THEN CONTINUE...
8539
8540
8541  024264                    SUB1C   SPAD (2),BREG,NOP
8542          000032            MICPC=MICPC+1
8543  024264  060342            .WORD   .$.
8544  024266                    BR      6$$
8545          000033            MICPC=MICPC+1
8546  024266  101035            .WORD   .$.
8547  024270                    SBR     1$
8548          000034            MICPC=MICPC+1
8549  024270  100406            .WORD   .$.
8550  024272
8551  024272                6$$:
8552          000035      3$:   MOVE    @ 0,BREG                            ;PREPARE TO CHECK IT.
8553  024272  000400            MICPC=MICPC+1
8554  024274                    .WORD   .$.
8555          000036            MOVE    @ 0,MLR                     ;LOAD MAR+LO.
8556  024274  010000            MICPC=MICPC+1
8557  024276                    .WORD   .$.
8558          000037            MOVE    @ 0,MPR                     ;LOAD MAR+HI.
8559  024276  004000            MICPC=MICPC+1
8560  024300                    .WORD   .$.
8561          000040            MOVE    BREG,SPAD (1)               ;SET THE PARALLEL ADDRESS.
8562  024300  063221            MICPC=MICPC+1
8563  024302                    .WORD   .$.
8564          000041            MOVE    BREG,SPAD (2)               ;SET THE PARALLEL ADDRESS.
8565  024302  063222            MICPC=MICPC+1
8566  024304                    .WORD   .$.
8567          000042      4$:   MOVE    MEM,BREG                    ;PUT "FOUND" IN BREG.
8568  024304  040620            MICPC=MICPC+1
8569  024306                    .WORD   .$.
8570                            $IFEQ   BREG,SPAD (1)   5$          ;BRANCH IF GOOD...
8571
```

```
8572  024306                  SUB2C   SPAD (1),BREG,NOP
8573  024310  000043          MICPC=MICPC+1
8574  024306  060361          .WORD   .$.
8575  024310                  BZ      5$
8576  024314  000044          MICPC=MICPC+1
8577  024310  101455          .WORD   .$.
8578  024312                  MOVE    MEM,OUT1 (CSR5)         ;
8579  024316  000045          MICPC=MICPC+1
8580  024312  041225          .WORD   .$.
8581  024314                  MOVE    SPAD (1),BREG           ;
8582  024320  000046          MICPC=MICPC+1
8583  024314  060601          .WORD   .$.
8584  024316                  MOVE    BREG,OUT1 (CSR4)        ;GOOD DATA...
8585  024322  000047          MICPC=MICPC+1
8586  024316  061224          .WORD   .$.
8587  024320                  MOVE    # 7,BREG                ;SET ERROR TYPE.
8588  024324  000050          MICPC=MICPC+1
8589  024320  000407          .WORD   .$.
8590  024322                  MOVE    BREG,OUT1 (CSR3)        ;
8591  024326  000051          MICPC=MICPC+1
8592  024322  061223          .WORD   .$.
8593  024324                  CALL    EROR1                   ;REPORT ERROR.
8594  024330  000052          MOVE    # (MICPC+3),BREG
8595  024324  000454          MICPC=MICPC+1
8596  024326                  .WORD   .$.
8597  024332  000053          SBR     ERROR1
8598  024326  104401          MICPC=MICPC+1
8599  024330                  .WORD   .$.
8600  024334  000054          SBR     4$                      ;LOOP ON ERROR.
8601  024330  100442          MICPC=MICPC+1
8602  024332                  .WORD   .$.
8603  024336  000055  5$:     CALL    SCP11                   ;SCOPE THE DATA??
8604  024332  000457          MOVE    # (MICPC+3),BREG
8605  024334                  MICPC=MICPC+1
8606  024340  000056          .WORD   .$.
8607  024334  104430          SBR     SCP11
8608  024336                  MICPC=MICPC+1
8609  024342  000057          .WORD   .$.
8610  024336  100442          SBR     4$                      ;YES, SCOPE IT.
8611  024340                  MICPC=MICPC+1
8612  024344  000060          .WORD   .$.
8613  024340  073061          $INC    SPAD (1), ,MLRLD        ;INCREMENT MEMORY ADDRESS.
8614  024342                  MICPC=MICPC+1
8615  024346  000061  .WORD   .$.!.DSP!MLRLD
8616  024342  067102          $ADC    SPAD (2), ,MPRLD        ;INCREMENT MEMORY ADDRESS.
8617  024344                  MICPC=MICPC+1
8618  024350  000062  .WORD   .$.!.DSP!MPRLD
8619  024344  000404          MOVE    # 4,BREG                        ;CHECK IF DONE??
8620  024346                  MICPC=MICPC+1
8621  024352  000063          .WORD   .$.
8622  024346                  $IFHI   BREG,SPAD (2)   4$      ;NO, THEN CONTINUE...
8623
8624
8625  024346                  SUB1C   SPAD (2),BREG,NOP
8626  024352  000063          MICPC=MICPC+1
8627  024346  060342          .WORD   .$.
```

```
024350                                          BC      65$
                 007064                         MICPC=MICPC+1
024350  101066                                  .WORD   .$.
024352                                          SBR     4$
                 000065                         MICPC=MICPC+1
024352  100442                                  .WORD   .$.
024354                                  65$:
024354                                          CALL    SCPE                    ;SCOPE THE TEST...
024354                                          MOVE    @ (MICPC+3),BREG
                 000066                         MICPC=MICPC+1
024354  000470                                  .WORD   .$.
024356                                          SBR     SCPE
                 000067                         MICPC=MICPC+1
024356  104454                                  .WORD   .$.
024360                                          SBR     6$                      ;DO THE NEXT ITERATION.
                 000070                         MICPC=MICPC+1
024360  100400                                  .WORD   .$.
024362                                  $MEM31
024362                                  $XZ


                                                ;*************************** TEST 41 ****************************
                                                ;* MAR TEST.
                                                ;* PERFORM DUAL ADDRESSING TEST.
                                                ;* USING MAR AUTO-INC FEATURE.
024362                                  $XZ
                                                ;:************************************************************

024362                                  $TSTN
                                                ;  TEST 41
                                                ;------------
024362  012737  000041  001202  TST41:  MOV     #41,$TSTNM              ; LOAD THE NO. OF THIS TEST
024370  012737  024544  001442          MOV     #TST42,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
024376  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
024402  024416                          MCT41
024404  104022                          ERROR   22                      ;TIME OUT ERROR...
024406  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
024412  000177  155024                  JMP     @NEXT                   ;GO TO NEXT TEST...
024416                                  MCT41:
024416                                  1$:     MOVE    @ 0,BREG                ;START WITH ZERO.
                 000000                         MICPC=MICPC+1
024416  000400                                  .WORD   .$.
024420                                          MOVE    BREG,SPAD (16)  ;FOR RETURN ADDRESS PURPOSES.
                 000001                         MICPC=MICPC+1
024420  063236                                  .WORD   .$.
024422                                          MOVE    @ 0,MLR                 ;LOAD MAR+LO.
                 000002                         MICPC=MICPC+1
024422  010000                                  .WORD   .$.
024424                                          MOVE    @ 0,MPR                 ;LOAD MAR+HI.
                 000003                         MICPC=MICPC+1
024424  004000                                  .WORD   .$.
024426                                          MOVE    BREG,SPAD (1)           ;SET PARALLEL ADDRESS.
                 000004                         MICPC=MICPC+1
024426  063221                                  .WORD   .$.
024430                                          MOVE    BREG,SPAD (2)           ;SET PARALLEL ADDRESS.
```

```
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # 4,BREG
                                MICPC=MICPC+1
                                .WORD   .$.
000005 063222           2$:     MOVE    SPAD (1),MEM,MARINC     ;LOAD DATA IN MEMORY.
                                MICPC=MICPC+1
                                .WORD   .$.
                                $INC    SPAD (1)                ;UPDATE ADDRESS.
                                MICPC=MICPC+1
                                .$.!.DSP
                                $ADC    SPAD (2)                ;UPDATE ADDRESS.
                                MICPC=MICPC+1
                                .DSP
                                II      BREG,SPAD (2)   2$      ;BRANCH IF NOT DONE...


                                C       SPAD (2),BREG,NOP
                                C=MICPC+1
                                D       .$.
                                        64$
                                C=MICPC+1
                                D       .$.
                                        2$
                                C=MICPC+1
                                .WORD   .$.

                                MOVE    # 0,BREG                ;START WITH 0 AGAIN.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # 0,MLR                 ;LOAD MAR<-LO.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # 0,MPR                 ;LOAD MAR<-HI.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    BREG,SPAD (1)           ;SET PARALLEL ADDRESS.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    BREG,SPAD (2)           ;SET PARALLEL ADDRESS.
                                MICPC=MICPC+1
                                .WORD   .$.
000022 063222           3$:     MOVE    MEM,BREG,MARINC         ;GET THE CONTENTS OF MEMORY
                                MICPC=MICPC+1
                                .WORD   .$.
                                $IFEQ   BREG,SPAD (1)   4$      ;BRANCH IF GOOD.


                                SUB2C   SPAD (1),BREG,NOP
                                MICPC=MICPC+1
                                .WORD   .$.
                                BZ      4$
                                MICPC=MICPC+1
                                .WORD   .$.
000025 101436                   MOVE    MEM,OUT1 (CSRS)         ;GOOD DATA.
                                MICPC=MICPC+1
```

```
024470 041225                   .WORD   .$
024472                          MOVE    SPAD (1),BREG           ;BAD DATA.
000026                          MICPC=MICPC+1
024472 060601                   .WORD   .$
024474                          MOVE    BREG,OUT1 (CSR4)        ;
000027                          MICPC=MICPC+1
024474 061224                   .WORD   .$
024476                          MOVE    # 10,BREG               ;SET THE ERROR TYPE.
000030                          MICPC=MICPC+1
024476 000410                   .WORD   .$
024500                          MOVE    BREG,OUT1 (CSR3)        ;
000031                          MICPC=MICPC+1
024500 061223                   .WORD   .$
024502                          CALL    ERROR                   ;REPORT ERROR.
                                MOVE    # (MICPC+3),BREG
000032                          MICPC=MICPC+1
024502 000434                   .WORD   .$
024504                          BBR     ERROR
000033                          MICPC=MICPC+1
024504 104400                   .WORD   .$
024506                          MOVE    SPAD (1),MLR    ;LOOP ON ERROR.
000034                          MICPC=MICPC+1
024506 070201                   .WORD   .$
024510                          BBR     3$                      ;LOOP ON ERROR.
000035                          MICPC=MICPC+1
024510 100422                   .WORD   .$
024512                  4$:     CALL    SCP1                    ;SCOPE THE DATA...
024512                          MOVE    # (MICPC+3),BREG
000036                          MICPC=MICPC+1
024512 000440                   .WORD   .$
024514                          BBR     SCP1
000037                          MICPC=MICPC+1
024514 104427                   .WORD   .$
024516                          MOVE    SPAD (1),MLR            ;RESTORE POINTER...
000040                          MICPC=MICPC+1
024516 070201                   .WORD   .$
024520                          BBR     3$                      ;LOOP ON DATA...
000041                          MICPC=MICPC+1
024520 100422                   .WORD   .$
024522                          SINC    SPAD (1)                        ;UPDATE ADDRESS.
000042                          MICPC=MICPC+1
024522 063061           .WORD   .$.!.DSP
024524                          SADC    SPAD (2)                        ;UPDATE ADDRESS.
000043                          MICPC=MICPC+1
024524 063102           .WORD   .$.!.DSP
024526                          MOVE    # 4,BREG                        ;CHECK IF DONE.
000044                          MICPC=MICPC+1
024526 000404                   .WORD   .$
024530                          SIFHI   BREG,SPAD (2)   3$      ;IF NOT THEN CONTINUE...


024530                          SUB1C   SPAD (2),BREG,NOP
000045                          MICPC=MICPC+1
024530 060342                   .WORD   .$
024532                          BC      65$
000046                          MICPC=MICPC+1
```

```
024532  101050                          .WORD   .$.
024534                                  $BR     .$.
        000047                          MICPC=MICPC+1
024534  100422                          .WORD   .$.
                                65$:
                                        CALL    SCPE                    ;SCOPE THE TEST.
                                        MOVE    # <MICPC+3>,BREG
        000050                          MICPC=MICPC+1
024536  000452                          .WORD   .$.
024540                                  $BR     SCPE
        000051                          MICPC=MICPC+1
024540  104454                          .WORD   .$.
024542                                  $BR     1$                      ;DO THE NEXT ITERATION...
        000052                          MICPC=MICPC+1
024542  100400                          .WORD   .$.
024544                          SALUTO
024544                          $XZ


                                        ;************************** TEST 42 **************************
                                        ;*ALU C BIT TEST
                                        ;*TEST THAT ADD OF 377 AND 1 WILL SET TAG CBIT.
                                        ;*THEN CHECK IF C BIT CLEARS
024544                          $XZ
                                        ;:**********************************************************

024544                          $TSTN
                                        ;  TEST 42
                                        ;--------------
024544  012737  000042  001202  TST42:  MOV     #42,$TSTNM              ; LOAD THE NO. OF THIS TEST
024552  012737  024700  001442          MOV     #TST43,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
024560  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
024564  024600                          MCT42
024566  104022                          ERROR   22                      ;TIME OUT ERROR...
024570  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
024574  000177  154642                  JMP     @NEXT                   ;GO TO NEXT TEST...
024600                          MCT42:
024600                          1$:     MOVE    # 0,BREG         ;SET TO CLEAR REQUIRED LOCATIONS.
        000000                          MICPC=MICPC+1
024600  000400                          .WORD   .$.
024602                                  MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES.
        000001                          MICPC=MICPC+1
024602  063236                          .WORD   .$.
024604                                  MOVE    # 0,MLR         ;
        000002                          MICPC=MICPC+1
024604  010000                          .WORD   .$.
024606                                  MOVE    # 0,MPR         ;
        000003                          MICPC=MICPC+1
024606  004000                          .WORD   .$.
024610                                  MOVE    BREG,SPAD <0>   ;CLEAR SPAD 0.
        000004                          MICPC=MICPC+1
024610  063220                          .WORD   .$.
024612                                  MOVE    BREG,SPAD <1>   ;CLEAR SPAD 1.
        000005                          MICPC=MICPC+1
024612  063221                          .WORD   .$.
```

```
024614                  MOVE    BREG,SPAD (2)   ;CLEAR SPAD 2.
                        MICPC=MICPC+1
000006                  .WORD   .$.
024614 063222           MOVE    # 377,MEM       ;
024616                  MICPC=MICPC+1
                        .WORD   .$.
000007                  MOVE    MEM,SPAD (1)    ;LOAD 377 IN SPAD 1.
024616 002777           MICPC=MICPC+1
024620                  .WORD   .$.
                        $INC    SPAD (1)                ;ADD 1 TO IT.
000010                  MICPC=MICPC+1
024620 043221           .WORD   .$.!.DSP
024622                  SADC    SPAD (2)                ;GET THE CARRY BIT..
                        MICPC=MICPC+1
000011                  .WORD   .$.!.DSP
024622 063061   .WORD   MOVE    SPAD (2),BREG   ;IS CARRY SET??
024624                  MICPC=MICPC+1
                        .WORD   .$.
000012                  BBO     2$              ;YES SCOPE THE TEST DATA.
024624 063102   .WORD   MICPC=MICPC+1
024626                  .WORD   .$.
                6$:     ;MOVE                   ;GOOD DATA.
000013                  MOVE    # 21,MEM                ;TYPE OF ERROR.
024626 060602           MICPC=MICPC+1
024630                  .WORD   .$.
                        MOVE    MEM,OUT1 (CSR3)         ;
000014                  MICPC=MICPC+1
024630 102022           .WORD   .$.
024632                  CALL    EROR1           ;REPORT ERROR...
                        MOVE    # (MICPC+3),BREG
024632                  MICPC=MICPC+1
                        .WORD   .$.
000015                  SBR     EROR1
024632 002421           MICPC=MICPC+1
024634                  .WORD   .$.
                        SBR     1$                      ;LOOP ON ERROR...
000016                  MICPC=MICPC+1
024634 041223           .WORD   .$.
                2$:     CALL    SCP11           ;SCOPE THE DATA??
024636                  MOVE    # (MICPC+3),BREG
                        MICPC=MICPC+1
024636                  .WORD   .$.
                        SBR     SCP11
000017                  MICPC=MICPC+1
024636 000421           .WORD   .$.
                        SBR     1$                      ;YES, DO IT...
024640                  MICPC=MICPC+1
                        .WORD   .$.
000020                  SDEC    SPAD (2)                ;RESET THE SPAD  (2).
024640 104401   .WORD   MICPC=MICPC+1
024642                  .WORD   .$.!.DSP
                        MOVE    # 377,MEM       ;LOAD 377 IN MEMORY.
000021                  MICPC=MICPC+1
024642 100400           .WORD   .$.
                        MOVE    MEM,SPAD (1)    ;LOAD IN SPAD.  (1).
024644                  MICPC=MICPC+1

000022
024644 000424
024646

000023
024646 104430
024650

000024
024650 100400
024652

000025
024652 063162
024654

000026
024654 002777
024656

000027
```

```
 8908  024656  043221                .WORD   .$.
 8909  024660                        $INC    SPAD <1>                ;ADD 1 TO IT.
 8910          000030                MICPC=MICPC+1
 8911  024660  063061        .WORD   .$.!.DSP
 8912  024662                        $DEC    SPAD <1>                ;SUBTRACT 1 OUT OF IT.
 8913          000031                MICPC=MICPC+1
 8914  024662  063161        .WORD   .$.!.DSP
 8915  024664                        $ADC    SPAD <0>        ;GET THE CARRY BIT.
 8916          000032                MICPC=MICPC+1
 8917  024664  063100        .WORD   .$.!.DSP
 8918  024666                4$:     MOVE    SPAD <0>,BREG   ;WAS IT RESET?
 8919          000033                MICPC=MICPC+1
 8920  024666  060600                .WORD   .$.
 8921  024670                        BBO     6$              ;NO, REPORT ERROR.
 8922          000034                MICPC=MICPC+1
 8923  024670  102015                .WORD   .$.
 8924  024672                        CALL    SCPE            ;SCOPE THE TEST.
 8925  024672                        MOVE    # <MICPC+3>,BREG
 8926          000035                MICPC=MICPC+1
 8927  024672  000437                .WORD   .$.
 8928  024674                        SBR     SCPE
 8929          000036                MICPC=MICPC+1
 8930  024674  104454                .WORD   .$.
 8931  024676                        SBR     1$              ;DO THE NEXT ITERATION.
 8932          000037                MICPC=MICPC+1
 8933  024676  100400                .WORD   .$.
 8934  024700                $ALUT1  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


                                ;************************** 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..
 8945  024700                $XZ
                                ;:*********************************************************************
 8947  024700                $TSTN
                                ;  TEST 43
                                ;-------------
 8951  024700  012737  000043  001202  TST43:  MOV     #43,$TSTNM              ; LOAD THE NO. OF THIS TEST
 8952  024706  012737  025226  001442          MOV     #TST44,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
 8954  024714  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
 8955  024720  024734                          MCT43
 8956  024722  104022                          ERROR   22                      ;TIME OUT ERROR...
 8957  024724  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
 8958  024730  000177  154506                  JMP     @NEXT                   ;GO TO NEXT TEST...
 8959  024734
 8960  024734                MCT43:
                             1$:        MOVE    # 0,MLR          ;SET MAR<-LO.
 8961          000000                  MICPC=MICPC+1
 8962  024734  010000                  .WORD   .$.
 8963  024736                          MOVE    # 0,MPR          ;SET MAR<-HI.
```

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

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

```
9075  025050  016652          .WORD   .$
9076  025052                  MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
9077                  000047  MICPC=MICPC+1
9078  025052  016652          .WORD   .$.
9079  025054                  MOVE    # 252,MEM MARINC        ;RESULT WITH C BIT SET.
9080                  000050  MICPC=MICPC+1
9081  025054  016652          .WORD   .$,
9082  025056                  MOVE    # 7,SPAD <7>    ;SET THE COUNT.
9083                  000051  MICPC=MICPC+1
9084  025056  003007          .WORD   .$.
9085
9086  025060                  MOVE    # 0,MLR          ;MAR+0.
9087                  000052  MICPC=MICPC+1
9088  025060  010000          .WORD   .$.
9089
9090  025062          2$:     MOVE    # 0,BREG                  ;
9091                  000053  MICPC=MICPC+1
9092  025062  000400          .WORD   .$.
9093  025064                  $ADD    SPAD <1>,BREG   ;CLEAR C BIT.
9094                  000054  MICPC=MICPC+1
9095  025064  060401          .WORD   .$.!.DD
9096  025066                  MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
9097                  000055  MICPC=MICPC+1
9098  025066  057220          .WORD   .$.
9099  025070                  MOVE    SPAD <0>,BR.SP,MARINC   ;
9100                  000056  MICPC=MICPC+1
9101  025070  077600          .WORD   .$.
9102  025072                  $IFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
9103
9104
9105
9106  025072                  SUB2C   SPAD <0>,MEM,NOP
9107                  000057  MICPC=MICPC+1
9108  025072  040360          .WORD   .$.
9109  025074                  BZ      3$
9110                  000060  MICPC=MICPC+1
9111  025074  101473          .WORD   .$.
9112  025076                  MOVE    MEM,OUT1 <4>    ;GOOD DATA
9113                  000061  MICPC=MICPC+1
9114  025076  041224          .WORD   .$.
9115  025100                  MOVE    BREG,OUT1 <5>   ;BAD DATA.
9116                  000062  MICPC=MICPC+1
9117  025100  061225          .WORD   .$.
9118  025102                  MOVE    # 15,BREG       ;SET TYPE OF ERROR.
9119                  000063  MICPC=MICPC+1
9120  025102  000415          .WORD   .$.
9121  025104                  MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
9122                  000064  MICPC=MICPC+1
9123  025104  061223          .WORD   .$.
9124  025106                  MOVE    # 11,BREG       ;LOAD FUNCTION CODE...
9125                  000065  MICPC=MICPC+1
9126  025106  000411          .WORD   .$.
9127  025110                  MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
9128                  000066  MICPC=MICPC+1
9129  025110  061227          .WORD   .$.
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   .$.
9134  025114                            SBR     ERROR
9135            000070                  MICPC=MICPC+1
9136  025114    104400                  .WORD   .$.
9137  025116                            MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
9138            000071                  MICPC=MICPC+1
9139  025116    070204                  .WORD   .$.
9140  025120                            SBR     2$              ;LOOP ON ERROR...
9141            000072                  MICPC=MICPC+1
9142  025120    100453                  .WORD   .$.
9143  025122                    3$:     CALL    SCP1
9144  025122                            MOVE    # <MICPC+3>,BREG
9145            000073                  MICPC=MICPC+1
9146  025122    000475                  .WORD   .$.
9147  025124                            SBR     SCP1
9148            000074                  MICPC=MICPC+1
9149  025124    104427                  .WORD   .$.
9150  025126                            MOVE    SPAD <4>,MLR    ;
9151            000075                  MICPC=MICPC+1
9152  025126    070204                  .WORD   .$.
9153  025130                            SBR     2$              ; SCOPE THE DATA....
9154            000076                  MICPC=MICPC+1
9155  025130    100453                  .WORD   .$.
9156  025132                    6$:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
9157            000077                  MICPC=MICPC+1
9158  025132    070204                  .WORD   .$.
9159  025134                            MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
9160            000100                  MICPC=MICPC+1
9161  025134    057220                  .WORD   .$.
9162  025136                            MOVE    # 377,BREG      ;
9163            000101                  MICPC=MICPC+1
9164  025136    000777                  .WORD   .$.
9165  025140                            $ADD    SPAD <2>,BREG   ;SET C BIT...
9166            000102                  MICPC=MICPC+1
9167  025140    060402                  .WORD   .$.!.DD
9168  025142                            MOVE    SPAD <0>,BR.SP,MARINC           ;SP 0 & BR = SEL B & SEL A
9169            000103                  MICPC=MICPC+1
9170  025142    077600                  .WORD   .$.
9171  025144                            MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
9172            000104                  MICPC=MICPC+1
9173  025144    057223                  .WORD   .$.
9174  025146                            $IFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...
9175
9176
9177  025146                            SUB2C   SPAD <0>,MEM,NOP
9178            000105                  MICPC=MICPC+1
9179  025146    040360                  .WORD   .$.
9180  025150                            BZ      9$
9181            000106                  MICPC=MICPC+1
9182  025150    101521                  .WORD   .$.
9183  025152                            MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
9184            000107                  MICPC=MICPC+1
9185  025152    041224                  .WORD   .$.
9186  025154                            MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
9187            000110                  MICPC=MICPC+1
```

```
                                .WORD   .$.
                                MOVE    # 23,BREG               ;ERROR TYPE...
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    BREG,OUT1 <CSR3>        ;
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # 11,BREG               ;ALU FUNCTION CODE.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
                                MICPC=MICPC+1
                                .WORD   .$.
                                CALL    ERROR   ;REPORT ERROR...
                                MOVE    # <MICPC+3>,BREG
                                MICPC=MICPC+1
                                .WORD   .$.
                                SBR     ERROR
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER.
                                MICPC=MICPC+1
                                .WORD   .$.
                                SBR     6$                      ;LOOP ON ERROR...
                                MICPC=MICPC+1
                                .WORD   .$.
                        9$:     CALL    SCP1            ;SCOPE THE ERROR...
                                MOVE    # <MICPC+3>,BREG
                                MICPC=MICPC+1
                                .WORD   .$.
                                SBR     SCP1
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    SPAD <4>,MLR    ;RESTORE DATA POINTER...
                                MICPC=MICPC+1
                                .WORD   .$.
                                SBR     6$              ;SCOPE THE DATA...
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
                                MICPC=MICPC+1
                                .WORD   .$.
                                SADD    BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
                                .WORD   .$.!MARINC!.DO
                                SDEC    SPAD <7>                ;IS IT DONE??
                                MICPC=MICPC+1
                        .WORD   .$.!.DSP
                                BZ      4$              ;YES, SCOPE THE TEST.
                                MICPC=MICPC+1
                                .WORD   .$.
                                SBR     2$              ;DO, THE NEXT.
                                MICPC=MICPC+1
                                .WORD   .$.
                        4$:     CALL    SCPE            ;SCOPE THE TEST...
                                MOVE    # <MICPC+3>,BREG
                                MICPC=MICPC+1
```

```
                                                .WORD   .$
                                                SBR     SCPE
                                                MICPC=MICPC+1
                                                .WORD   .$.
                                                SBR     .$                      ;DO THE NEXT ITERATION...
                                                MICPC=MICPC+1
                                                .WORD   .$.
                                        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>
                                        $XZ


                                                ;*************************** 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..
025226                                  $XZ
                                                ;;************************************************************************

025226                                  $TSTN
                                                ;  TEST 44
                                                ;------------
025226  012737  000044  001202  TST44:  MOV     #44,$TSTNM              ; LOAD THE NO. OF THIS TEST
025234  012737  025554  001442          MOV     #TST45,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
025242  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
025246  025262                          MCT44
025250  104022                          ERROR   22                      ;TIME OUT ERROR...
025252  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
025256  000177  154160                  JMP     @NEXT                   ;GO TO NEXT TEST...
025262                          MCT44:
025262                          15:     MOVE    # 0,MLR          ;SET MAR<-LO.
000000                                  MICPC=MICPC+1
025262  010000                          .WORD   .$.
025264                                  MOVE    # 0,MPR          ;SET MAR<-HI.
000001                                  MICPC=MICPC+1
025264  004000                          .WORD   .$.
025266                                  MOVE    # 0,BREG                 ;
000002                                  MICPC=MICPC+1
025266  000400                          .WORD   .$.
025270                                  MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS...
000003                                  MICPC=MICPC+1
025270  063236                          .WORD   .$.
025272                                  MOVE    BREG,SPAD <0>   ;
000004                                  MICPC=MICPC+1
025272  063220                          .WORD   .$.
025274                                  MOVE    BREG,SPAD <1>   ;
000005                                  MICPC=MICPC+1
025274  063221                          .WORD   .$.
025276                                  MOVE    BREG,SPAD <2>   ;
000006                                  MICPC=MICPC+1
025276  063222                          .WORD   .$.
025300                                  $DEC    SPAD <2>                ;
000007                                  MICPC=MICPC+1
```

```
                        .WORD   $,!.DSP
                                MOVE    BREG,SPAD <4>           ;
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # -1,MEM MARINC
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # -1,MEM MARINC
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # 0,MEM MARINC
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # -1,MEM MARINC
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
```

```
                                                MICPC=MICPC+1
                                                .WORD   .S.
025346  000032  016525                          MOVE    # -1,MEM MARINC
025350                                          MICPC=MICPC+1
                                                .WORD   .S.
025352  000033  016777                          MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
025354                                          MICPC=MICPC+1
                                                .WORD   .S.
025356  000034  016777                          MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
025360                                          MICPC=MICPC+1
                                                .WORD   .S.
025362  000035  016652                          MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
025364                                          MICPC=MICPC+1
                                                .WORD   .S.
025366  000036  016525                          MOVE    # 252,MEM MARINC
025370                                          MICPC=MICPC+1
                                                .WORD   .S.
025372  000037  016652                          MOVE    # 252,MEM MARINC         ;RESULT WITH C BIT SET.
025374                                          MICPC=MICPC+1
                                                .WORD   .S.
025376  000040  016652                          MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
025400                                          MICPC=MICPC+1
                                                .WORD   .S.
025402  000041  016525                          MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY
025404                                          MICPC=MICPC+1
                                                .WORD   .S.
025406  000042  016652                          MOVE    # 125,MEM MARINC
025410                                          MICPC=MICPC+1
                                                .WORD   .S.
025412  000043  016525                          MOVE    # 125,MEM MARINC         ;RESULT WITH C BIT SET.
025414                                          MICPC=MICPC+1
                                                .WORD   .S.
025416  000044  016652                          MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
025420                                          MICPC=MICPC+1
                                                .WORD   .S.
025422  000045  016652                          MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
025424                                          MICPC=MICPC+1
                                                .WORD   .S.
025426  000046  016652                          MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
025430                                          MICPC=MICPC+1
                                                .WORD   .S.
025432  000047  016777                          MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
025434                                          MICPC=MICPC+1
                                                .WORD   .S.
025436  000050  016777                          MOVE    # 7,SPAD <7>     ;SET THE COUNT.
025440                                          MICPC=MICPC+1
                                                .WORD   .S.
9401    000051
9402    025404  003007
9403
9404    025406                                  MOVE    # 0,MLR          ;MAR=0.
9405            000052                          MICPC=MICPC+1
9406    025406  010000                          .WORD   .S.
9407
9408    025410                          2$:     MOVE    # 0,BREG                 ;
9409            000053                          MICPC=MICPC+1
9410    025410  000400                          .WORD   .S.
9411    025412                                  $ADD    SPAD <1>,BREG   ;CLEAR C BIT.
```

```
9412                    000054          MICPC=MICPC+1
9413    025412  060401                  .WORD   .$.!.DD
9414    025414                          MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
9415                    000055          MICPC=MICPC+1
9416    025414  057220                  .WORD   .$.
9417    025416                          LOAM    MEM,SPAD <0>,BR.SP,MARINC       ;SPAD  <0>:=$FUNC.
9418                    000056          MICPC=MICPC+1
9419    025416  057640                  .WORD   .$.
9420    025420                          $IFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
9421
9422
9423    025420                          SUB2C   SPAD <0>,MEM,NOP
9424                    000057          MICPC=MICPC+1
9425    025420  040060                  .WORD   .$.
9426    025422                          BZ      3$
9427                    000060          MICPC=MICPC+1
9428    025422  101473                  .WORD   .$.
9429    025424                          MOVE    MEM,OUT1 <4>    ;GOOD DATA
9430                    000061          MICPC=MICPC+1
9431    025424  041224                  .WORD   .$.
9432    025426                          MOVE    BREG,OUT1 <5>   ;BAD DATA.
9433                    000062          MICPC=MICPC+1
9434    025426  051225                  .WORD   .$.
9435    025430                          MOVE    # 15,BREG       ;SET TYPE OF ERROR.
9436                    000063          MICPC=MICPC+1
9437    025430  000415                  .WORD   .$.
9438    025432                          MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
9439                    000064          MICPC=MICPC+1
9440    025432  051227                  .WORD   .$.
9441    025434                          MOVE    # 12,BREG       ;LOAD FUNCTION CODE...
9442                    000065          MICPC=MICPC+1
9443    025434  000412                  .WORD   .$.
9444    025436                          MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
9445                    000066          MICPC=MICPC+1
9446    025436  051227                  .WORD   .$.
9447    025440                          CALL    ERROR           ;ALU A OR NOTB ERROR...
9448    025442                          MOVE    # <MICPC+3>,BREG
9449                    000067          MICPC=MICPC+1
9450    025442  000471                  .WORD   .$.
9451    025444                          BR      ERROR
9452                    000070          MICPC=MICPC+1
9453    025444  104400                  .WORD   .$.
9454    025446                          MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
9455                    000071          MICPC=MICPC+1
9456    025446  070004                  .WORD   .$.
9457    025450                          BR      .$.             ;LOOP ON ERROR...
9458                    000072          MICPC=MICPC+1
9459    025450  104453                  .WORD   .$.
9460    025452          3$:             CALL    SCP1
9461    025454                          MOVE    # <MICPC+3>,BREG
9462                    000073          MICPC=MICPC+1
9463    025454  000475                  .WORD   .$.
9464    025456                          BR      SCP1
9465                    000074          MICPC=MICPC+1
9466    025456  104427                  .WORD   .$.
9467    025460                          MOVE    SPAD <4>,MLR    ;
```

```
                    000075                  MICPC=MICPC+1
025454  070204      .WORD   .$.
025456              SBR     25              ; SCOPE THE DATA....
                    000076                  MICPC=MICPC+1
025460  100453      .WORD   .$.
025460          65: MOVE    SPAD (4),MLR    ;RESET DATA POINTER...
                    000077                  MICPC=MICPC+1
025462  070204      .WORD   .$.
025464              MOVE    MEM,SPAD (0),MARINC     ;GET FIRST OPRAND...
                    000100                  MICPC=MICPC+1
025466  057220      .WORD   .$.
025470              MOVE    # 377,BREG      ;
                    000101                  MICPC=MICPC+1
025472  000777      .WORD   .$.
025474              SADD    SPAD (2),BREG   ;SET C BIT...
                    000102                  MICPC=MICPC+1
025476  060402      .WORD   .$.!.DD
025500              LORM    MEM,SPAD (0),BR.SP,MARINC       ;SP 0 & BR = A OR NOTB
                    000103                  MICPC=MICPC+1
025502  057640      .WORD   .$.
025504              MOVE    MEM,SPAD (3),MARINC     ;DUMMY INSTR, TO MARINC.
                    000104                  MICPC=MICPC+1
025506  057223      .WORD   .$.
025510              SIFEQ   MEM,SPAD (0)    95      ;BR IF GOOD...


025512                      SUB2C   SPAD (0),MEM,NOP
                    000105  MICPC=MICPC+1
025514  040360      .WORD   .$.
025516              BZ      95
                    000106  MICPC=MICPC+1
025520  101521      .WORD   .$.
025522              MOVE    MEM,OUT1 (CSR4) ;GOOD DATA.
                    000107  MICPC=MICPC+1
025524  041224      .WORD   .$.
025526              MOVE    BREG,OUT1 (CSR5)        ;BAD DATA.
                    000110  MICPC=MICPC+1
025530  061225      .WORD   .$.
025532              MOVE    # 23,BREG               ;ERROR TYPE...
                    000111  MICPC=MICPC+1
025534  000423      .WORD   .$.
025536              MOVE    BREG,OUT1 (CSR3)        ;
                    000112  MICPC=MICPC+1
025540  061223      .WORD   .$.
025542              MOVE    # 12,BREG               ;ALU FUNCTION CODE.
                    000113  MICPC=MICPC+1
025544  000412      .WORD   .$.
025546              MOVE    BREG,OUT1 (CSR7)        ;LOAD IT...
                    000114  MICPC=MICPC+1
025550  061227      .WORD   .$.
025552              CALL    ERROR   ;REPORT ERROR...
025554              MOVE    # (MICPC+3),BREG
                    000115  MICPC=MICPC+1
025556  000517      .WORD   .$.
025560              SBR     ERROR
                    000116  MICPC=MICPC+1
```

```
9524  025516  104400                    .WORD   .$.
9525  025520                            MOVE    SPAD (4),MLR            ;RESTORE DATA POINTER.
                000117                  MICPC=MICPC+1
9526  025520  070204                    .WORD   .$.
9527  025522                            SBR     6$                      ;LOOP ON ERROR...
                000120                  MICPC=MICPC+1
9528  025522  100477                    .WORD   .$.
9529  025524                    95:     CALL    SCP1                    ;SCOPE THE ERROR...
                000121                  MOVE    # <MICPC+3>,BREG
9530  025524  000523                    MICPC=MICPC+1
9531  025526                            .WORD   .$.
                000122                  SBR     SCP1
9532  025526  104427                    MICPC=MICPC+1
9533  025530                            .WORD   .$.
                000123                  MOVE    SPAD (4),MLR    ;RESTORE DATA POINTER...
9540  025530  070204                    MICPC=MICPC+1
9541  025532                            .WORD   .$.
                000124                  SBR     6$                      ;SCOPE THE DATA...
9542                                    MICPC=MICPC+1
9543  025532  100477                    .WORD   .$.
9544  025534                            MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
                000125                  MICPC=MICPC+1
9545  025534  000404                    .WORD   .$.
9546  025536                            SADD    BREG,SPAD (4),MARINC    ;ALSO DATA POINTER.
9547  025536  077004                    .WORD   .$.!MARINC!.DO
9548  025540                            SDEC    SPAD (7)                ;IS IT DONE??
                000126                  MICPC=MICPC+1
9549  025540  063167          .WORD     .$.!.DSP
9550  025542                            BZ      4$                      ;YES, SCOPE THE TEST.
                000127                  MICPC=MICPC+1
9551  025542  101532                    .WORD   .$.
9552  025544                            SBR     2$                      ;DO, THE NEXT.
                000130                  MICPC=MICPC+1
9553  025544  100453                    .WORD   .$.
9554  025546                    4$:     CALL    SCPE                    ;SCOPE THE TEST...
                000131                  MOVE    # <MICPC+3>,BREG
9555  025546  000533                    MICPC=MICPC+1
9556  025550                            .WORD   .$.
                000132                  SBR     SCPE
9557  025550  104454                    MICPC=MICPC+1
9560  025552                            .WORD   .$.
                000133                  SBR     1$                      ;DO THE NEXT ITERATION...
9561  025552  100400                    MICPC=MICPC+1
9562  025554                    $ALUT1  .WORD   .$.
9563  025554                    $XZ     0,<A AND B>,AND,0,0,0,0,0,0,0,-1,-1,125,125,0,0,0,0,252,252,<A AND B>,1,13
9564
9565
9566                                    ;************************** TEST 45 ****************************
9567                                    ;*ALU TEST
9570                                    ;*TEST OF ALU FUNCTION A AND B WITH C BIT CLEARED.
9571                                    ;*TEST OF ALU FUNCTION A AND B WITH C BIT SET.
9572                                    ;*ALU FUNCTION (A AND B)
9573                                    ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
9574                                    ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
9575
9576
9577
9600
9601  025554                    $XZ
```

```
                                        ;;*****************************************************************
                                $TSTN
                                        ;  TEST 45
                                        ;-------------
025554 012737 000045 001202     TST45:  MOV     #45,$TSTNM              ; LOAD THE NO. OF THIS TEST
025562 012737 026102 001442             MOV     #TST46,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
025570 004737 035536                    JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
025574 025610                           MCT45
025576 104022                           ERROR   22                      ;TIME OUT ERROR...
025600 012706 001200                    MOV     #STACK,SP               ;RESET STACK...
025604 000177 153632                    JMP     @NEXT                   ;GO TO NEXT TEST...
025610                          MCT45:
025610                          1$:     MOVE    # 0,MLR          ;SET MAR<-LO.
        000000                          MICPC=MICPC+1
025610 010000                           .WORD   .$.
025612                                  MOVE    # 0,MPR          ;SET MAR<-HI.
        000001                          MICPC=MICPC+1
025612 004000                           .WORD   .$.
025614                                  MOVE    # 0,BREG                 ;
        000002                          MICPC=MICPC+1
025614 000400                           .WORD   .$.
025616                                  MOVE    BREG,SPAD (16)          ;FOR RETURN ADDRESS...
        000003                          MICPC=MICPC+1
025616 063236                           .WORD   .$.
025620                                  MOVE    BREG,SPAD (0)   ;
        000004                          MICPC=MICPC+1
025620 063220                           .WORD   .$.
025622                                  MOVE    BREG,SPAD (1)   ;
        000005                          MICPC=MICPC+1
025622 063221                           .WORD   .$.
025624                                  MOVE    BREG,SPAD (2)   ;
        000006                          MICPC=MICPC+1
025624 063222                           .WORD   .$.
025626                                  SDEC    SPAD (2)                ;
        000007                          MICPC=MICPC+1
025626 063162                   .WORD   .$.!.DSP
                                        MOVE    BREG,SPAD (4)           ;
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    # 0,MEM MARINC
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
                                        MICPC=MICPC+1
                                        .WORD   .$.
```

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

```
9692  025710  016400                .WORD   .$.
9693  025712                        MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
9694          000041                MICPC=MICPC+1
9695  025712  016525                .WORD   .$.
9696  025714                        MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY
9697          000042                MICPC=MICPC+1
9698  025714  016652                .WORD   .$.
9699  025716                        MOVE    # 0,MEM MARINC
9700          000043                MICPC=MICPC+1
9701  025716  016400                .WORD   .$.
9702  025720                        MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
9703          000044                MICPC=MICPC+1
9704  025720  016400                .WORD   .$.
9705  025722                        MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
9706          000045                MICPC=MICPC+1
9707  025722  016652                .WORD   .$.
9708  025724                        MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
9709          000046                MICPC=MICPC+1
9710  025724  016652                .WORD   .$.
9711  025726                        MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
9712          000047                MICPC=MICPC+1
9713  025726  016652                .WORD   .$.
9714  025730                        MOVE    # 252,MEM MARINC        ;RESULT WITH C BIT SET.
9715          000050                MICPC=MICPC+1
9716  025730  016652                .WORD   .$.
9717  025732                        MOVE    # 7,SPAD <7>    ;SET THE COUNT.
9718          000051                MICPC=MICPC+1
9719  025732  003007                .WORD   .$.
9720
9721  025734                        MOVE    # 0,MLR         ;MAR+0.
9722          000052                MICPC=MICPC+1
9723  025734  010000                .WORD   .$.
9724
9725  025736                2$:     MOVE    # 0,BREG                 ;
9726          000053                MICPC=MICPC+1
9727  025736  000400                .WORD   .$.
9728  025740                        $ADD    SPAD <1>,BREG   ;CLEAR C BIT.
9729          000054                MICPC=MICPC+1
9730  025740  060401                .WORD   .$.!.DD
9731  025742                        MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
9732          000055                MICPC=MICPC+1
9733  025742  057220                .WORD   .$.
9734  025744                        AND     MEM,SPAD <0>,BR.SP,MARINC       ;SPAD  <0>:=$FUNC.
9735          000056                MICPC=MICPC+1
9736  025744  057660                .WORD   .$.
9737  025746                        $IFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
9738
9739
9740  025746                        SUB2C   SPAD <0>,MEM,NOP
9741          000057                MICPC=MICPC+1
9742  025746  040360                .WORD   .$.
9743  025750                        BZ      3$
9744          000060                MICPC=MICPC+1
9745  025750  101473                .WORD   .$.
9746  025752                        MOVE    MEM,OUT1 <4>    ;GOOD DATA
9747          000061                MICPC=MICPC+1
```

```
9748  025752  041224                          .WORD   .$.
9749  025754                                  MOVE    BREG,OUT1 <5>   ;BAD DATA.
9750          000062                          MICPC=MICPC+1
9751  025754  061225                          .WORD   .$.
9752  025756                                  MOVE    # 15,BREG       ;SET TYPE OF ERROR.
9753          000063                          MICPC=MICPC+1
9754  025756  000415                          .WORD   .$.
9755  025760                                  MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
9756          000064                          MICPC=MICPC+1
9757  025760  061223                          .WORD   .$.
9758  025762                                  MOVE    # 13,BREG       ;LOAD FUNCTION CODE...
9759          000065                          MICPC=MICPC+1
9760  025762  000413                          .WORD   .$.
9761  025764                                  MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
9762          000066                          MICPC=MICPC+1
9763  025764  061227                          .WORD   .$.
9764  025766                                  CALL    ERROR           ;ALU A AND B ERROR...
9765  025766                                  MOVE    # <MICPC+3>,BREG
9766          000067                          MICPC=MICPC+1
9767  025766  000471                          .WORD   .$.
9768  025770                                  SBR     ERROR
9769          000070                          MICPC=MICPC+1
9770  025770  104400                          .WORD   .$.
9771  025772                                  MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
9772          000071                          MICPC=MICPC+1
9773  025772  070204                          .WORD   .$.
9774  025774                                  SBR     2$              ;LOOP ON ERROR...
9775          000072                          MICPC=MICPC+1
9776  025774  100453                          .WORD   .$.
9777  025776                          3$:     CALL    SCP1
9778  025776                                  MOVE    # <MICPC+3>,BREG
9779          000073                          MICPC=MICPC+1
9780  025776  000475                          .WORD   .$.
9781  026000                                  SBR     SCP1
9782          000074                          MICPC=MICPC+1
9783  026000  104427                          .WORD   .$.
9784  026002                                  MOVE    SPAD <4>,MLR    ;
9785          000075                          MICPC=MICPC+1
9786  026002  070204                          .WORD   .$.
9787  026004                                  SBR     2$              ; SCOPE THE DATA....
9788          000076                          MICPC=MICPC+1
9789  026004  100453                          .WORD   .$.
9790  026006                          6$:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
9791          000077                          MICPC=MICPC+1
9792  026006  070204                          .WORD   .$.
9793  026010                                  MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
9794          000100                          MICPC=MICPC+1
9795  026010  057220                          .WORD   .$.
9796  026012                                  MOVE    # 377,BREG      ;
9797          000101                          MICPC=MICPC+1
9798  026012  000777                          .WORD   .$.
9799  026014                                  SADD    SPAD <2>,BREG   ;SET C BIT...
9800          000102                          MICPC=MICPC+1
9801  026014  060402                          .WORD   .$.!.DD
9802  026016                                  AND     MEM,SPAD <0>,BR.SP,MARINC       ;SP 0 & BR = A AND B
9803          000103                          MICPC=MICPC+1
```

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

```
026060 100477                      .WORD   .$
026062                             MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
       000125                      MICPC=MICPC+1
026062 000404                      .WORD   .$.
026064                             SPOD    BREG,SPOD (4),MARINC    ;ALSO DATA POINTER.
026064 077004                      .WORD   .$.!MARINC!.DD
026066                             SDEC    SPOD (7)                ;IS IT DONE??
       000126                      MICPC=MICPC+1
026066 063167              .WORD   .$.!.DSP
026070                             BZ      45                      ;YES, SCOPE THE TEST.
       000127                      MICPC=MICPC+1
026070 101532                      .WORD   .$.
026072                             SBR     2$                      ;DO, THE NEXT.
       000130                      MICPC=MICPC+1
026072 100453                      .WORD   .$.
026074                     45:     CALL    SCPE                    ;SCOPE THE TEST...
026074                             MOVE    # <MICPC+3>,BREG
       000131                      MICPC=MICPC+1
026074 000533                      .WORD   .$.
026076                             SBR     SCPE
       000132                      MICPC=MICPC+1
026076 104454                      .WORD   .$.
026100                             SBR     1$                      ;DO THE NEXT ITERATION...
       000133                      MICPC=MICPC+1
026100 100400                      .WORD   .$.
026102                     $ALUT1  0,<A OR B>,OR,0,0,-1,-1,-1,-1,-1,-1,125,125,-1,-1,-1,-1,252,252,<A OR B>,1,14
026102                     $XZ


                                   ;************************** 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..
026102                     $XZ
                                   ;;*****************************************************************
026102                     $TSTN
                                   ;  TEST 46
                                   ;--------------
026102 012737 000046 001202 TST46: MOV     #46,$TSTNM              ; LOAD THE NO. OF THIS TEST
026110 012737 026430 001442        MOV     #TST47,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                           ;R1 CONTAINS BASE KMC11 ADDRESS
026116 004737 035536               JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
026122 026136                      MCT46
026124 104022                      ERROR   22                      ;TIME OUT ERROR...
026126 012706 001200               MOV     #STACK,SP               ;RESET STACK...
026132 000177 153304               JMP     @NEXT                   ;GO TO NEXT TEST...
026136                     MCT46:
026136                     1$:     MOVE    # 0,MLR          ;SET MAR<-LO.
       000000                      MICPC=MICPC+1
026136 010000                      .WORD   .$.
026140                             MOVE    # 0,MPR          ;SET MAR<-HI.
       000001                      MICPC=MICPC+1
```

```
9916  026140  004000                    .WORD   .$
9917  026142                            MOVE    # 0,BREG                ;
9918          000002                    MICPC=MICPC+1
9919  026142  000400                    .WORD   .$
9920  026144                            MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS...
9921          000003                    MICPC=MICPC+1
9922  026144  063236                    .WORD   .$
9923  026146                            MOVE    BREG,SPAD <0>   ;
9924          000004                    MICPC=MICPC+1
9925  026146  063220                    .WORD   .$
9926  026150                            MOVE    BREG,SPAD <1>   ;
9927          000005                    MICPC=MICPC+1
9928  026150  063221                    .WORD   .$
9929  026152                            MOVE    BREG,SPAD <2>   ;
9930          000006                    MICPC=MICPC+1
9931  026152  063222                    .WORD   .$
9932  026154                            SDEC    SPAD <2>                ;
9933          000007                    MICPC=MICPC+1
9934  026154  063162            .WORD   .$.!.DSP
9935  026156                            MOVE    BREG,SPAD <4>           ;
9936          000010                    MICPC=MICPC+1
9937  026156  063224                    .WORD   .$
9938  026160                            MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
9939          000011                    MICPC=MICPC+1
9940  026160  016400                    .WORD   .$
9941  026162                            MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
9942          000012                    MICPC=MICPC+1
9943  026162  016400                    .WORD   .$
9944  026164                            MOVE    # 0,MEM MARINC
9945          000013                    MICPC=MICPC+1
9946  026164  016400                    .WORD   .$
9947  026166                            MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
9948          000014                    MICPC=MICPC+1
9949  026166  016401                    .WORD   .$
9950  026170                            MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
9951          000015                    MICPC=MICPC+1
9952  026170  016777                    .WORD   .$
9953  026172                            MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
9954          000016                    MICPC=MICPC+1
9955  026172  016400                    .WORD   .$
9956  026174                            MOVE    # -1,MEM MARINC
9957          000017                    MICPC=MICPC+1
9958  026174  016777                    .WORD   .$
9959  026176                            MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
9960          000020                    MICPC=MICPC+1
9961  026176  016777                    .WORD   .$
9962  026200                            MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
9963          000021                    MICPC=MICPC+1
9964  026200  016400                    .WORD   .$
9965  026202                            MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
9966          000022                    MICPC=MICPC+1
9967  026202  016777                    .WORD   .$
9968  026204                            MOVE    # -1,MEM MARINC
9969          000023                    MICPC=MICPC+1
9970  026204  016777                    .WORD   .$
9971  026206                            MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
```

```
                 007024
026206  016777
026210
                 000025
026210  016777
026212
                 000026
026212  016777
026214
                 000027
026214  016777
026216
                 000030
026216  016777
026220
                 000031
026220  016525
026222
                 000032
026222  016525
026224
                 000033
026224  016525
026226
                 000034
026226  016525
026230
                 000035
026230  016652
026232
                 000036
026232  016525
026234
                 000037
026234  016777
026236
                 000040
026236  016777
026240
                 000041
026240  016525
026242
                 000042
026242  016652
026244
                 000043
026244  016777
026246
                 000044
026246  016777
026250
                 000045
026250  016652
026252
                 000046
026252  016652

MICPC=MICPC+1
.WORD   .$.
MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
MICPC=MICPC+1
.WORD   .$.
MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .$.
MOVE    # -1,MEM MARINC
MICPC=MICPC+1
.WORD   .$.
MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 125,MEM MARINC          ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 125,MEM MARINC          ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 125,MEM MARINC
MICPC=MICPC+1
.WORD   .$.
MOVE    # 125,MEM MARINC          ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 252,MEM MARINC          ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 125,MEM MARINC          ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .$.
MOVE    # -1,MEM MARINC
MICPC=MICPC+1
.WORD   .$.
MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 125,MEM MARINC          ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 252,MEM MARINC          ;LOAD THE DATA IN MEMORY
MICPC=MICPC+1
.WORD   .$.
MOVE    # -1,MEM MARINC
MICPC=MICPC+1
.WORD   .$.
MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 252,MEM MARINC          ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 252,MEM MARINC          ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .$.
```

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

```
10084  026314  000471                .WORD   .$.
10085  026316                        $BR     ERROR
10086          000070                MICPC=MICPC+1
10087  026316  104400                .WORD   .$.
10088  026320                        MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
10089          000071                MICPC=MICPC+1
10090  026320  070204                .WORD   .$.
10091  026322                        $BR     2$              ;LOOP ON ERROR...
10092          000072                MICPC=MICPC+1
10093  026322  100453                .WORD   .$.
10094  026324                3$:     CALL    SCP1
10095  026324                        MOVE    # <MICPC+3>,BREG
10096          000073                MICPC=MICPC+1
10097  026324  000475                .WORD   .$.
10098  026326                        $BR     SCP1
10099          000074                MICPC=MICPC+1
10100  026326  104427                .WORD   .$.
10101  026330                        MOVE    SPAD <4>,MLR    ;
10102          000075                MICPC=MICPC+1
10103  026330  070204                .WORD   .$.
10104  026332                        $BR     2$              ; SCOPE THE DATA....
10105          000076                MICPC=MICPC+1
10106  026332  100453                .WORD   .$.
10107  026334                6$:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
10108          000077                MICPC=MICPC+1
10109  026334  070204                .WORD   .$.
10110  026336                        MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
10111          000100                MICPC=MICPC+1
10112  026336  057220                .WORD   .$.
10113  026340                        MOVE    # 377,BREG      ;
10114          000101                MICPC=MICPC+1
10115  026340  000777                .WORD   .$.
10116  026342                        SADD    SPAD <2>,BREG   ;SET C BIT...
10117          000102                MICPC=MICPC+1
10118  026342  060402                .WORD   .$.!.DD
10119  026344                        OR      MEM,SPAD <0>,BR.SP,MARINC       ;SP 0 & BR = A OR B
10120          000103                MICPC=MICPC+1
10121  026344  057700                .WORD   .$.
10122  026346                        MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MAR.INC.
10123          000104                MICPC=MICPC+1
10124  026346  057223                .WORD   .$.
10125  026350                        $IFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...
10126
10127
10128  026350                        SUB2C   SPAD <0>,MEM,NOP
10129          000105                MICPC=MICPC+1
10130  026350  040360                .WORD   .$.
10131  026352                        BZ      9$
10132          000106                MICPC=MICPC+1
10133  026352  101521                .WORD   .$.
10134  026354                        MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
10135          000107                MICPC=MICPC+1
10136  026354  041224                .WORD   .$.
10137  026356                        MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
10138          000110                MICPC=MICPC+1
10139  026356  061225                .WORD   .$.
```

```
10140 026360                    MOVE    # 23,BREG                ;ERROR TYPE...
10141         000111            MICPC=MICPC+1
10142 026360  000423            .WORD   .$.
10143 026362                    MOVE    BREG,OUT1 (CSR3)        ;
10144         000112            MICPC=MICPC+1
10145 026362  061223            .WORD   .$.
10146 026364                    MOVE    # 14,BREG               ;ALU FUNCTION CODE.
10147         000113            MICPC=MICPC+1
10148 026364  000414            .WORD   .$.
10149 026366                    MOVE    BREG,OUT1 (CSR7)        ;LOAD IT...
10150         000114            MICPC=MICPC+1
10151 026366  061227            .WORD   .$.
10152 026370                    CALL    ERROR   ;REPORT ERROR...
10153 026370                    MOVE    # (MICPC+3),BREG
10154         000115            MICPC=MICPC+1
10155 026370  000517            .WORD   .$.
10156 026372                    SBR     ERROR
10157         000116            MICPC=MICPC+1
10158 026372  104400            .WORD   .$.
10159 026374                    MOVE    SPAD (4),MLR            ;RESTORE DATA POINTER.
10160         000117            MICPC=MICPC+1
10161 026374  070204            .WORD   .$.
10162 026376                    SBR     6$                      ;LOOP ON ERROR...
10163         000120            MICPC=MICPC+1
10164 026376  100477            .WORD   .$.
10165 026400            9$:     CALL    SCP1            ;SCOPE THE ERROR...
10166 026400                    MOVE    # (MICPC+3),BREG
10167         000121            MICPC=MICPC+1
10168 026400  000523            .WORD   .$.
10169 026402                    SBR     SCP1
10170         000122            MICPC=MICPC+1
10171 026402  104427            .WORD   .$.
10172 026404                    MOVE    SPAD (4),MLR    ;RESTORE DATA POINTER...
10173         000123            MICPC=MICPC+1
10174 026404  070204            .WORD   .$.
10175 026406                    SBR     6$              ;SCOPE THE DATA...
10176         000124            MICPC=MICPC+1
10177 026406  100477            .WORD   .$.
10178 026410                    MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
10179         000125            MICPC=MICPC+1
10180 026410  000404            .WORD   .$.
10181 026412                    SADD    BREG,SPAD (4),MARINC    ;ALSO DATA POINTER.
10182 026412  077004            .WORD   .$.!MARINC!.DD
10183 026414                    SDEC    SPAD (7)                ;IS IT DONE??
10184         000126            MICPC=MICPC+1
10185 026414  063167    .WORD   .$.!.DSP
10186 026416                    BZ      4$                      ;YES, SCOPE THE TEST.
10187         000127            MICPC=MICPC+1
10188 026416  101532            .WORD   .$.
10189 026420                    SBR     2$                      ,DD, THE NEXT.
10190         000130            MICPC=MICPC+1
10191 026420  100453            .WORD   .$.
10192 026422            4$:     CALL    SCPE            ;SCOPE THE TEST...
10193 026422                    MOVE    # (MICPC+3),BREG
10194         000131            MICPC=MICPC+1
10195 026422  000533            .WORD   .$.
```

```
10196  026424                                  SBR     SCPE
10197          000132                          MICPC=MICPC+1
10198  026424  104454                          .WORD   .$.
10199  026426                                  SBR     I$              ;DO THE NEXT ITERATION...
10200          000133                          MICPC=MICPC+1
10201  026426  100400                          .WORD   .$.
10202  026430                          $ALUT1  0,<A XOR B>,$XOR,0,0,-1,-1,-1,-1,0,0,0,0,-1,-1,-1,-1,0,0,<A XOR B>,1,15
10203  026430                          $XZ
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                          $XZ
10214                                           ;:********************************************************************
10215
10216  026430                          $TSTN
10217                                   ;  TEST 47
10218                                   ;-------------
10219  026430  012737  000047  001202  TST47:  MOV     #47,$TSTNM              ; LOAD THE NO. OF THIS TEST
10220  026436  012737  026756  001442          MOV     #TST50,NEXT             ; POINT TO THE START OF NEXT TEST.
10221                                                          ;R1 CONTAINS BASE KMC11 ADDRESS
10222  026444  004737  035E36                  JSR     PC,LDVRWT               ;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                          MCT47:
10228  026464                          1$:     MOVE    # 0,MLR          ;SET MAR<LO.
10229          000000                          MICPC=MICPC+1
10230  026464  010000                          .WORD   .$.
10231  026466                                  MOVE    # 0,MPR          ;SET MAR<HI.
10232          000001                          MICPC=MICPC+1
10233  026466  004000                          .WORD   .$.
10234  026470                                  MOVE    # 0,BREG                 ;
10235          000002                          MICPC=MICPC+1
10236  026470  000400                          .WORD   .$.
10237  026472                                  MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS...
10238          000003                          MICPC=MICPC+1
10239  026472  063236                          .WORD   .$.
10240  026474                                  MOVE    BREG,SPAD <0>   ;
10241          000004                          MICPC=MICPC+1
10242  026474  063220                          .WORD   .$.
10243  026476                                  MOVE    BREG,SPAD <1>   ;
10244          000005                          MICPC=MICPC+1
10245  026476  063221                          .WORD   .$.
10246  026500                                  MOVE    BREG,SPAD <2>   ;
10247          000006                          MICPC=MICPC+1
10248  026500  063222                          .WORD   .$.
10249  026502                                  SDEC    SPAD <2>                ;
10250          000007                          MICPC=MICPC+1
10251  026502  063162                  .WORD   .$.!.DSP
```

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

```
10308  026550  016525          .WORD   .$.
10309  026552                  MOVE    # 0,MEM MARINC
10310          000033          MICPC=MICPC+1
10311  026552  016400          .WORD   .$.
10312  026554                  MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
10313          000034          MICPC=MICPC+1
10314  026554  016400          .WORD   .$.
10315  026556                  MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10316          000035          MICPC=MICPC+1
10317  026556  016652          .WORD   .$.
10320  026560                  MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10319          000036          MICPC=MICPC+1
10320  026560  016525          .WORD   .$.
10321  026562                  MOVE    # -1,MEM MARINC
10322          000037          MICPC=MICPC+1
10323  026562  016777          .WORD   .$.
10324  026564                  MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
10325          000040          MICPC=MICPC+1
10326  026564  016777          .WORD   .$.
10327  026566                  MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10328          000041          MICPC=MICPC+1
10329  026566  016525          .WORD   .$.
10330  026570                  MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY
10331          000042          MICPC=MICPC+1
10332  026570  016652          .WORD   .$.
10333  026572                  MOVE    # -1,MEM MARINC
10334          000043          MICPC=MICPC+1
10335  026572  016777          .WORD   .$.
10336  026574                  MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
10337          000044          MICPC=MICPC+1
10338  026574  016777          .WORD   .$.
10339  026576                  MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10340          000045          MICPC=MICPC+1
10341  026576  016652          .WORD   .$.
10342  026600                  MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10343          000046          MICPC=MICPC+1
10344  026600  016652          .WORD   .$.
10345  026602                  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
10346          000047          MICPC=MICPC+1
10347  026602  016400          .WORD   .$.
10348  026604                  MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
10349          000050          MICPC=MICPC+1
10350  026604  016400          .WORD   .$.
10351  026606                  MOVE    # 7,SPAD <7>    ;SET THE COUNT.
10352          000051          MICPC=MICPC+1
10353  026606  003007          .WORD   .$.
10354
10355  026610                  MOVE    # 0,MLR         ;MAR←0.
10356          000052          MICPC=MICPC+1
10357  026610  010000          .WORD   .$.
10358
10359  026612          2$:     MOVE    # 0,BREG                 ;
10360          000053          MICPC=MICPC+1
10361  026612  000400          .WORD   .$.
10362  026614                  SADD    SPAD <1>,BREG  ;CLEAR C BIT.
10363          000054          MICPC=MICPC+1
```

```
10364  026614  060401          .WORD   .$.!.DD
10365  026616                  MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
10366          000055          MICPC=MICPC+1
10367  026616  057220          .WORD   .$.
10370  026620                  $XOR    MEM,SPAD <0>,BR.SP,MARINC       ;SPAD  <0>:=$FUNC.
10371          000056          MICPC=MICPC+1
10372  026620  057720          .WORD   .$.
10373  026622                  $IFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
10374
10375  026622                  SUB2C   SPAD <0>,MEM,NOP
10376          000057          MICPC=MICPC+1
10377  026622  040360          .WORD   .$.
10400  026624                  BZ      3$
10401          000060          MICPC=MICPC+1
10402  026624  101473          .WORD   .$.
10403  026626                  MOVE    MEM,OUT1 <4>    ;GOOD DATA
10404          000061          MICPC=MICPC+1
10405  026626  041224          .WORD   .$.
10406  026630                  MOVE    BREG,OUT1 <5>   ;BAD DATA.
10407          000062          MICPC=MICPC+1
10410  026630  061225          .WORD   .$.
10411  026632                  MOVE    # 15,BREG       ;SET TYPE OF ERROR.
10412          000063          MICPC=MICPC+1
10413  026632  000415          .WORD   .$.
10414  026634                  MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
10415          000064          MICPC=MICPC+1
10416  026634  061223          .WORD   .$.
10417  026636                  MOVE    # 15,BREG       ;LOAD FUNCTION CODE...
10420          000065          MICPC=MICPC+1
10421  026636  000415          .WORD   .$.
10422  026640                  MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
10423          000066          MICPC=MICPC+1
10424  026640  061227          .WORD   .$.
10425  026642                  CALL    EROR            ;ALU A XOR B ERROR...
10426  026642                  MOVE    # <MICPC+3>,BREG
10427          000067          MICPC=MICPC+1
10430  026642  000471          .WORD   .$.
10431  026644                  $BR     EROR
10432          000070          MICPC=MICPC+1
10433  026644  104400          .WORD   .$.
10434  026646                  MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
10435          000071          MICPC=MICPC+1
10436  026646  070204          .WORD   .$.
10437  026650                  $BR     2$              ;LOOP ON ERROR...
10440          000072          MICPC=MICPC+1
10441  026650  100453          .WORD   .$.
10442  026652          3$:     CALL    SCP1
10443  026652                  MOVE    # <MICPC+3>,BREG
10444          000073          MICPC=MICPC+1
10445  026652  000475          .WORD   .$.
10446  026654                  $BR     SCP1
10447          000074          MICPC=MICPC+1
10450  026654  104427          .WORD   .$.
10451  026656                  MOVE    SPAD <4>,MLR    ;
10452          000075          MICPC=MICPC+1
```

```
10420  026656  070204                    .WORD   .$.
10421  026660                            $BR     2$                  ; SCOPE THE DATA....
10422          000076                    MICPC=MICPC+1
10423  026660  100453                    .WORD   .$.
10424  026662                    6$:     MOVE    SPAD (4),MLR        ;RESET DATA POINTER...
10425          000077                    MICPC=MICPC+1
10426  026662  070204                    .WORD   .$.
10427  026664                            MOVE    MEM,SPAD (0),MARINC     ;GET FIRST OPRAND...
10428          000100                    MICPC=MICPC+1
10429  026664  057220                    .WORD   .$.
10430  026666                            MOVE    # 377,BREG          ;
10431          000101                    MICPC=MICPC+1
10432  026666  000777                    .WORD   .$.
10433  026670                            $AOD    SPAD (2),BREG       ;SET C BIT...
10434          000102                    MICPC=MICPC+1
10435  026670  060402                    .WORD   .$.!.DD
10436  026672                            $XOR    MEM,SPAD (0),BR.SP,MARINC       ;SP 0 & BR = A XOR B
10437          000103                    MICPC=MICPC+1
10438  026672  057720                    .WORD   .$.
10439  026674                            MOVE    MEM,SPAD (3),MARINC     ;DUMMY INSTR, TO MARINC.
10440          000104                    MICPC=MICPC+1
10441  026674  057223                    .WORD   .$.
10442  026676                            $IFEQ   MEM,SPAD (0)    9$      ;BR IF GOOD...
10443
10444
10445  026676                            SUB2C   SPAD (0),MEM,NOP
10446          000105                    MICPC=MICPC+1
10447  026676  040360                    .WORD   .$.
10448  026700                            BZ      9$
10449          000106                    MICPC=MICPC+1
10450  026700  101521                    .WORD   .$.
10451  026702                            MOVE    MEM,OUT1 (CSR4) ;GOOD DATA.
10452          000107                    MICPC=MICPC+1
10453  026702  041224                    .WORD   .$.
10454  026704                            MOVE    BREG,OUT1 (CSR5)        ;BAD DATA.
10455          000110                    MICPC=MICPC+1
10456  026704  061225                    .WORD   .$.
10457  026706                            MOVE    # 23,BREG               ;ERROR TYPE...
10458          000111                    MICPC=MICPC+1
10459  026706  000423                    .WORD   .$.
10460  026710                            MOVE    BREG,OUT1 (CSR3)        ;
10461          000112                    MICPC=MICPC+1
10462  026710  061223                    .WORD   .$.
10463  026712                            MOVE    # 15,BREG               ;ALU FUNCTION CODE.
10464          000113                    MICPC=MICPC+1
10465  026712  000415                    .WORD   .$.
10466  026714                            MOVE    BREG,OUT1 (CSR7)        ;LOAD IT...
10467          000114                    MICPC=MICPC+1
10468  026714  061227                    .WORD   .$.
10469  026716                            CALL    EROR    ;REPORT ERROR...
10470  026716                            MOVE    # (MICPC+3),BREG
10471          000115                    MICPC=MICPC+1
10472  026716  000517                    .WORD   .$.
10473  026720                            $BR     EROR
10474          000116                    MICPC=MICPC+1
10475  026720  104400                    .WORD   .$.
```

```
10476  026722                          MOVE    SPAD (4),MLR            ;RESTORE DATA POINTER.
10477          000117                  MICPC=MICPC+1
10478  026722  070204                  .WORD   .$.
10479  026724                          SBR     6$                      ;LOOP ON ERROR...
10480          000120                  MICPC=MICPC+1
10481  026724  100477                  .WORD   .$.
10482  026726                  9$:     CALL    $CP1                    ;SCOPE THE ERROR...
10483  026726                          MOVE    # <MICPC+3>,BREG
10484          000121                  MICPC=MICPC+1
10485  026726  000523                  .WORD   .$.
10486  026730                          SBR     $CP1
10487          000122                  MICPC=MICPC+1
10488  026730  104427                  .WORD   .$.
10489  026732                          MOVE    SPAD (4),MLR    ;RESTORE DATA POINTER...
10490          000123                  MICPC=MICPC+1
10491  026732  070204                  .WORD   .$.
10492  026734                          SBR     6$                      ;SCOPE THE DATA...
10493          000124                  MICPC=MICPC+1
10494  026734  100477                  .WORD   .$.
10495  026736                          MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
10496          000125                  MICPC=MICPC+1
10497  026736  000404                  .WORD   .$.
10498  026740                          SADD    BREG,SPAD (4),MARINC    ;ALSO DATA POINTER.
10499  026740  077004                  .WORD   .$.!MARINC!.DD
10500  026742                          SDEC    SPAD (7)                ;IS IT DONE??
10501          000126                  MICPC=MICPC+1
10502  026742  063167          .WORD   .$.!.DSP
10503  026744                          BZ      4$                      ;YES, SCOPE THE TEST.
10504          000127                  MICPC=MICPC+1
10505  026744  101532                  .WORD   .$.
10506  026746                          SBR     2$                      ;DO, THE NEXT.
10507          000130                  MICPC=MICPC+1
10508  026746  100453                  .WORD   .$.
10509  026750                  4$:     CALL    $CPE                    ;SCOPE THE TEST...
10510  026750                          MOVE    # <MICPC+3>,BREG
10511          000131                  MICPC=MICPC+1
10512  026750  000533                  .WORD   .$.
10513  026752                          SBR     $CPE
10514          000132                  MICPC=MICPC+1
10515  026752  104454                  .WORD   .$.
10516  026754                          SBR     1$                      ;DO THE NEXT ITERATION...
10517          000133                  MICPC=MICPC+1
10518  026754  100400                  .WORD   .$.
10519  026756                  SALUT1  0,<ADD>,$ADD,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                                   $TSTN
10534                                                    ;  TEST 50
10535                                                    ;--------------
10536  026756  012737  000050  001202  TST50:  MOV     #50,$TSTNM          ; LOAD THE NO. OF THIS TEST
10537  026764  012737  027304  001442           MOV     #TST51,NEXT         ; POINT TO THE START OF NEXT TEST.
10538                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
10539  026772  004737  035536                   JSR     PC,LDVRWT           ;LOAD-VERIFY-WAIT.
10540  026776  027012                           MCT50
10541  027000  104022                           ERROR   22                  ;TIME OUT ERROR...
10542  027002  012706  001200                   MOV     #STACK,SP           ;RESET STACK...
10543  027006  000177  152430                   JMP     @NEXT               ;GO TO NEXT TEST...
10544  027012                           MCT50:
10545  027012                           1$:     MOVE    # 0,MLR             ;SET MAR<-LO.
10546          000000                           MICPC=MICPC+1
10547  027012  010000                           .WORD   .$.
10548  027014                                   MOVE    # 0,MPR             ;SET MAR<-HI.
10549          000001                           MICPC=MICPC+1
10550  027014  004000                           .WORD   .$.
10551  027016                                   MOVE    # 0,BREG                     ;
10552          000002                           MICPC=MICPC+1
10553  027016  000400                           .WORD   .$.
10554  027020                                   MOVE    BREG,SPAD <16>              ;FOR RETURN ADDRESS...
10555          000003                           MICPC=MICPC+1
10556  027020  063236                           .WORD   .$.
10557  027022                                   MOVE    BREG,SPAD <0>       ;
10558          000004                           MICPC=MICPC+1
10559  027022  063220                           .WORD   .$.
10560  027024                                   MOVE    BREG,SPAD <1>       ;
10561          000005                           MICPC=MICPC+1
10562  027024  063221                           .WORD   .$.
10563  027026                                   MOVE    BREG,SPAD <2>       ;
10564          000006                           MICPC=MICPC+1
10565  027026  063222                           .WORD   .$.
10566  027030                                   SDEC    SPAD <2>                    ;
10567          000007                           MICPC=MICPC+1
10568  027030  063162                   .WORD   .$.!.DSP
10569  027032                                   MOVE    BREG,SPAD <4>               ;
10570          000010                           MICPC=MICPC+1
10571  027032  063224                           .WORD   .$.
10572  027034                                   MOVE    # 0,MEM MARINC      ;LOAD THE DATA IN MEMORY.
10573          000011                           MICPC=MICPC+1
10574  027034  016400                           .WORD   .$.
10575  027036                                   MOVE    # 0,MEM MARINC      ;LOAD THE DATA IN MEMORY
10576          000012                           MICPC=MICPC+1
10577  027036  016400                           .WORD   .$.
10578  027040                                   MOVE    # 0,MEM MARINC
10579          000013                           MICPC=MICPC+1
10580  027040  016400                           .WORD   .$.
10581  027042                                   MOVE    # 0,MEM MARINC      ;RESULT WITH C BIT SET.
10582          000014                           MICPC=MICPC+1
10583  027042  016400                           .WORD   .$.
10584  027044                                   MOVE    # -1,MEM MARINC     ;LOAD THE DATA IN MEMORY.
10585          000015                           MICPC=MICPC+1
10586  027044  016777                           .WORD   .$.
10587  027046                                   MOVE    # 0,MEM MARINC      ;LOAD THE DATA IN MEMORY.
```

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

```
10644 027114
10645          000041
10646 027114   016525
10647 027116
10648          000042
10649 027116   016652
10650 027120
10651          000043
10652 027120   016777
10653 027122
10654          000044
10655 027122   016777
10656 027124
10657          000045
10658 027124   016652
10659 027126
10660          000046
10661 027126   016652
10662 027130
10663          000047
10664 027130   016524
10665 027132
10666          000050
10667 027132   016524
10668 027134
10669          000051
10670 027134   003007
10671
10672 027136
10673          000052
10674 027136   010000
10675
10676 027140
10677          000053
10678 027140   000400
10679 027142
10680          000054
10681 027142   060401
10682 027144
10683          000055
10684 027144   057220
10685 027146
10686          000056
10687 027146   057400
10688 027150
10689
10690
10691 027150
10692          000057
10693 027150   040360
10694 027152
10695          000060
10696 027152   101473
10697 027154
10698          000061
10699 027154   041224
```

```
        MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # -1,MEM MARINC
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 124,MEM MARINC         ;LOAD THE DATA IN MEMORY.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 124,MEM MARINC         ;RESULT WITH C BIT SET.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 7,SPAD <7>     ;SET THE COUNT.
        MICPC=MICPC+1
        .WORD   .$.

        MOVE    # 0,MLR          ;MAR+0.
        MICPC=MICPC+1
        .WORD   .$.

2$:     MOVE    # 0,BREG                  ;
        MICPC=MICPC+1
        .WORD   .$.
        $ADD    SPAD <1>,BREG   ;CLEAR C BIT.
        MICPC=MICPC+1
        .WORD   .$.!.DD
        MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
        MICPC=MICPC+1
        .WORD   .$.
        $ADD    MEM,SPAD <0>,BR.SP,MARINC       ;SPAD  <0>:=SFUNC.
        MICPC=MICPC+1
        .WORD   .$.
        $IFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.


        SUB2C   SPAD <0>,MEM,NOP
        MICPC=MICPC+1
        .WORD   .$.
        BZ      3$
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,OUT1 <4>    ;GOOD DATA
        MICPC=MICPC+1
        .WORD   .$.
```

```
10700  027156                          MOVE    BREG,OUT1 (5)   ;BAD DATA.
10701                  000062           MICPC=MICPC+1
10702  027156  061225                  .WORD   .$.
10703  027160                          MOVE    # 15,BREG       ;SET TYPE OF ERROR.
10704                  000063           MICPC=MICPC+1
10705  027160  000415                  .WORD   .$.
10706  027162                          MOVE    BREG,OUT1 (3)   ;SET TYPE OF ERROR.
10707                  000064           MICPC=MICPC+1
10710  027162  061223                  .WORD   .$.
10711  027164                          MOVE    # 00,BREG       ;LOAD FUNCTION CODE...
10712                  000065           MICPC=MICPC+1
10713  027164  000400                  .WORD   .$.
10714  027166                          MOVE    BREG,OUT1 (CSR7)        ;LOAD IT...
10715                  000066           MICPC=MICPC+1
10716  027166  061227                  .WORD   .$.
10717  027170                          CALL    EROR            ;ALU ADD ERROR...
10720                  000067           MOVE    # (MICPC+3),BREG
10721  027170  000471                  MICPC=MICPC+1
10722  027172                          .WORD   .$.
10723                  000070           SBR     EROR
10724  027172  104400                  MICPC=MICPC+1
10725  027174                          .WORD   .$.
10726                  000071           MOVE    SPAD (4),MLR    ;RESET DATA POINTER...
10727  027174  070204                  MICPC=MICPC+1
10730  027176                          .WORD   .$.
10731                  000072           SBR     2$              ;LOOP ON ERROR...
10732  027176  100453                  MICPC=MICPC+1
10733  027200                          .WORD   .$.
10734  027200          3$:             CALL    SCP1
10735                  000073           MOVE    # (MICPC+3),BREG
10736  027200  000475                  MICPC=MICPC+1
10737  027202                          .WORD   .$.
10740                  000074           SBR     SCP1
10741  027202  104427                  MICPC=MICPC+1
10742  027204                          .WORD   .$.
10743                  000075           MOVE    SPAD (4),MLR    ;
10744  027204  070204                  MICPC=MICPC+1
10745  027206                          .WORD   .$.
10746                  000076           SBR     2$              ; SCOPE THE DATA....
10747  027206  100453                  MICPC=MICPC+1
10750  027210                          .WORD   .$.
10751                  000077   6$:     MOVE    SPAD (4),MLR    ;RESET DATA POINTER...
10752  027210  070204                  MICPC=MICPC+1
10753  027212                          .WORD   .$.
10754                  000100           MOVE    MEM,SPAD (0),MARINC     ;GET FIRST OPRAND...
10755  027212  057220                  MICPC=MICPC+1
10756  027214                          .WORD   .$.
10757                  000101           MOVE    # 377,BREG      ;
10760  027214  000777                  MICPC=MICPC+1
10761  027216                          .WORD   .$.
10762                  000102           SADD    SPAD (2),BREG   ;SET C BIT...
10763  027216  060402                  MICPC=MICPC+1
10764  027220                          .WORD   .$.!.DD
10765                  000103           SADD    MEM,SPAD (0),BR.SP,MARINC       ;SP 0 & BR = ADD
10766  027220  057400                  MICPC=MICPC+1
                                        .WORD   .$.
```

```
10756  027222                  MOVE    MEM,SPAD (3),MARINC     ;DUMMY INSTR, TO MARINC.
10757          000104          MICPC=MICPC+1
10758  027222  057223          .WORD   .$.
10759  027224                  $IFEQ   MEM,SPAD (0)    9$      ;BR IF GOOD...
10760
10761
10762  027224                  SUB2C   SPAD (0),MEM,NOP
10763          000105          MICPC=MICPC+1
10764  027224  040360          .WORD   .$.
10765  027226                  BZ      9$
10766          000106          MICPC=MICPC+1
10767  027226  101521          .WORD   .$.
10768  027230                  MOVE    MEM,OUT1 (CSR4) ;GOOD DATA.
10769          000107          MICPC=MICPC+1
10770  027230  041224          .WORD   .$.
10771  027232                  MOVE    BREG,OUT1 (CSR5)        ;BAD DATA.
10772          000110          MICPC=MICPC+1
10773  027232  061225          .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   .$.
10786  027244                  CALL    ERROR   ;REPORT ERROR...
10787  027244                  MOVE    # (MICPC+3),BREG
10788          000115          MICPC=MICPC+1
10789  027244  000517          .WORD   .$.
10790  027246                  SBR     ERROR
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     6$                      ;LOOP ON ERROR...
10797          000120          MICPC=MICPC+1
10798  027252  100477          .WORD   .$.
10799  027254          9$:     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   .$.
10809  027262                  SBR     6$              ;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  000404                          .WORD   .$.
10815  027266                                  SADD    BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
10816  027266  077004                          .WORD   .$.!MARINC!.DO
10817  027270                                  SDEC    SPAD <7>                ;IS IT DONE??
10818          000126                          MICPC=MICPC+1
10819  027270  063167                  .WORD   .$.!.DSP
10820  027272                                  BZ      4$                      ;YES, SCOPE THE TEST.
10821          000127                          MICPC=MICPC+1
10822  027272  101532                          .WORD   .$.
10823  027274                                  SBR     2$                      ;DO, THE NEXT.
10824          000130                          MICPC=MICPC+1
10825  027274  100453                          .WORD   .$.
10826  027276                          4$:     CALL    SCPE                    ;SCOPE THE TEST...
10827  027276                                  MOVE    # <MICPC+3>,BREG
10828          000131                          MICPC=MICPC+1
10829  027276  001533                          .WORD   .$.
10830  027300                                  SBR     SCPE
10831          000132                          MICPC=MICPC+1
10832  027300  104454                          .WORD   .$.
10833  027302                                  SBR     1$                      ;DO THE NEXT ITERATION...
10834          000133                          MICPC=MICPC+1
10835  027302  100400                          .WORD   .$.
10836  027304                          $ALUT:  0,<2A W/C>,$ROL,0,1,376,-1,0,1,376,-1,252,253,124,125,252,253,124,125,<A PLUS A
10837  027304                          $XZ
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                          $XZ
10848                                          ;;*********************************************************************
10849  027304                          $TSTN
10850                                          ;  TEST 51
10851                                          ;-------------
10852  027304  012737  000051  001202  TST51:  MOV     #51,$TSTNM              ; LOAD THE NO. OF THIS TEST
10853  027312  012737  027632  001442          MOV     #TST52,NEXT             ; POINT TO THE START OF NEXT TEST.
10854                                                                  ;R1 CONTAINS BASE KMC11 ADDRESS
10855  027320  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
10856  027324  027340                          MCT51
10857  027326  104022                          ERROR   22                      ;TIME OUT ERROR...
10858  027330  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
10859  027334  000177  152102                  JMP     @NEXT                   ;GO TO NEXT TEST...
10860  027340                          MCT51:
10861  027340                          1$:     MOVE    # 0,MLR          ;SET MAR<LO.
10862          000000                          MICPC=MICPC+1
10863  027340  010000                          .WORD   .$.
10864  027342                                  MOVE    # 0,MPR          ;SET MAR<HI.
10865          000001                          MICPC=MICPC+1
10866  027342  004000                          .WORD   .$.
10867
```

```
10868 027344                    MOVE    # 0,BREG                ;
10869         000002            MICPC=MICPC+1
10870 027344  000400            .WORD   .$.
10871 027346                    MOVE    BREG,SPAD (16)          ;FOR RETURN ADDRESS...
10872         000003            MICPC=MICPC+1
10873 027346  063236            .WORD   .$.
10874 027350                    MOVE    BREG,SPAD (0)   ;
10875         000004            MICPC=MICPC+1
10876 027350  063220            .WORD   .$.
10877 027352                    MOVE    BREG,SPAD (1)   ;
10878         000005            MICPC=MICPC+1
10879 027352  063221            .WORD   .$.
10880 027354                    MOVE    BREG,SPAD (2)   ;
10881         000006            MICPC=MICPC+1
10882 027354  063222            .WORD   .$.
10883 027356                    SDEC    SPAD (2)                ;
10884         000007            MICPC=MICPC+1
10885 027356  063162    .WORD   .$.!.DSP
10886 027360                    MOVE    BREG,SPAD (4)           ;
10887         000010            MICPC=MICPC+1
10888 027360  063224            .WORD   .$.
10889 027362                    MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
10890         000011            MICPC=MICPC+1
10891 027362  016400            .WORD   .$.
10892 027364                    MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
10893         000012            MICPC=MICPC+1
10894 027364  016400            .WORD   .$.
10895 027366                    MOVE    # 0,MEM MARINC
10896         000013            MICPC=MICPC+1
10897 027366  016400            .WORD   .$.
10898 027370                    MOVE    # 1,MEM MARINC  ;RESULT WITH C BIT SET.
10899         000014            MICPC=MICPC+1
10900 027370  016401            .WORD   .$.
10901 027372                    MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
10902         000015            MICPC=MICPC+1
10903 027372  016777            .WORD   .$.
10904 027374                    MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
10905         000016            MICPC=MICPC+1
10906 027374  016400            .WORD   .$.
10907 027376                    MOVE    # 376,MEM MARINC
10908         000017            MICPC=MICPC+1
10909 027376  016776            .WORD   .$.
10910 027400                    MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
10911         000020            MICPC=MICPC+1
10912 027400  016777            .WORD   .$.
10913 027402                    MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
10914         000021            MICPC=MICPC+1
10915 027402  016400            .WORD   .$.
10916 027404                    MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
10917         000022            MICPC=MICPC+1
10918 027404  016777            .WORD   .$.
10919 027406                    MOVE    # 0,MEM MARINC
10920         000023            MICPC=MICPC+1
10921 027406  016400            .WORD   .$.
10922 027410                    MOVE    # 1,MEM MARINC  ;RESULT WITH C BIT SET.
10923         000024            MICPC=MICPC+1
```

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

```
10980           000047          MICPC=MICPC+1
10981  027456   016524          .WORD   .$.
10982  027460                   MOVE    # 125,MEM MARINC        ;RESULT WITH C BIT SET.
10983           000050          MICPC=MICPC+1
10984  027460   016525          .WORD   .$.
10985  027462                   MOVE    # 7,SPAD <7>    ;SET THE COUNT.
10986           000051          MICPC=MICPC+1
10987  027462   003007          .WORD   .$.
10988
10989  027464                   MOVE    # 0,MLR         ;MAR+0.
10990           000052          MICPC=MICPC+1
10991  027464   010000          .WORD   .$.
10992
10993  027466                   2$:     MOVE    # 0,BREG                 ;
10994           000053          MICPC=MICPC+1
10995  027466   000400          .WORD   .$.
10996  027470                   $ADD    SPAD <1>,BREG   ;CLEAR C BIT.
10997           000054          MICPC=MICPC+1
10998  027470   760401          .WORD   .$.!.DO
10999  027472                   MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
11000           000055          MICPC=MICPC+1
11001  027472   057220          .WORD   .$.
11002  027474                   $ROL    SPAD <0>,BR.SP,MARINC   ;
11003           000056          MICPC=MICPC+1
11004  027474   077540          .WORD   .$.
11005  027476                   $IFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
11006
11007
11008  027476                   SUB2C   SPAD <0>,MEM,NOP
11009           000057          MICPC=MICPC+1
11010  027476   040360          .WORD   .$.
11011  027500                   BZ      3$
11012           000060          MICPC=MICPC+1
11013  027500   101473          .WORD   .$.
11014  027502                   MOVE    MEM,OUT1 <4>    ;GOOD DATA
11015           000061          MICPC=MICPC+1
11016  027502   041224          .WORD   .$.
11017  027504                   MOVE    BREG,OUT1 <5>   ;BAD DATA.
11018           000062          MICPC=MICPC+1
11019  027504   061225          .WORD   .$.
11020  027506                   MOVE    # 15,BREG       ;SET TYPE OF ERROR.
11021           000063          MICPC=MICPC+1
11022  027506   000415          .WORD   .$.
11023  027510                   MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
11024           000064          MICPC=MICPC+1
11025  027510   061223          .WORD   .$.
11026  027512                   MOVE    # 06,BREG       ;LOAD FUNCTION CODE...
11027           000065          MICPC=MICPC+1
11028  027512   000406          .WORD   .$.
11029  027514                   MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
11030           000066          MICPC=MICPC+1
11031  027514   061227          .WORD   .$.
11032  027516                   CALL    ERROR           ;ALU 2A W/C ERROR...
11033  027516                   MOVE    # <MICPC+3>,BREG
11034           000067          MICPC=MICPC+1
11035  027516   000471          .WORD   .$.
```

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

```
11092                   000111          MICPC=MICPC+1
11093  027562  000423                   .WORD   .$.
11094  027564                           MOVE    BREG,OUT1 <CSR3>        ;
11095                   000112          MICPC=MICPC+1
11096  027564  061223                   .WORD   .$.
11097  027566                           MOVE    # 06,BREG               ;ALU FUNCTION CODE.
11098                   000113          MICPC=MICPC+1
11099  027566  000406                   .WORD   .$.
11100  027570                           MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
11101                   000114          MICPC=MICPC+1
11102  027570  061227                   .WORD   .$.
11103  027572                           CALL    ERROR   ;REPORT ERROR...
11104  027572                           MOVE    # <MICPC+3>,BREG
11105                   000115          MICPC=MICPC+1
11106  027572  000517                   .WORD   .$.
11107  027574                           SBR     ERROR
11108                   000116          MICPC=MICPC+1
11109  027574  104400                   .WORD   .$.
11110  027576                           MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER.
11111                   000117          MICPC=MICPC+1
11112  027576  070204                   .WORD   .$.
11113  027600                           SBR     6$                      ;LOOP ON ERROR...
11114                   000120          MICPC=MICPC+1
11115  027600  100477                   .WORD   .$.
11116  027602                   9$:     CALL    SCP1            ;SCOPE THE ERROR...
11117  027602                           MOVE    # <MICPC+3>,BREG
11118                   000121          MICPC=MICPC+1
11119  027602  000523                   .WORD   .$.
11120  027604                           SBR     SCP1
11121                   000122          MICPC=MICPC+1
11122  027604  104427                   .WORD   .$.
11123  027606                           MOVE    SPAD <4>,MLR    ;RESTORE DATA POINTER...
11124                   000123          MICPC=MICPC+1
11125  027606  070204                   .WORD   .$.
11126  027610                           SBR     6$                      ;SCOPE THE DATA...
11127                   000124          MICPC=MICPC+1
11128  027610  100477                   .WORD   .$.
11129  027612                           MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
11130                   000125          MICPC=MICPC+1
11131  027612  000404                   .WORD   .$.
11132  027614                           SADD    BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
11133  027614  077004                   .WORD   .$.!MARINC!.DO
11134  027616                           SDEC    SPAD <7>                ;IS IT DONE??
11135                   000126          MICPC=MICPC+1
11136  027616  063167           .WORD   .$.!.DSP
11137  027620                           BZ      4$                      ;YES, SCOPE THE TEST.
11138                   000127          MICPC=MICPC+1
11139  027620  101532                   .WORD   .$.
11140  027622                           SBR     2$                      ;DO, THE NEXT.
11141                   000130          MICPC=MICPC+1
11142  027622  100453                   .WORD   .$.
11143  027624                   4$:     CALL    SCPE                    ;SCOPE THE TEST...
11144  027624                           MOVE    # <MICPC+3>,BREG
11145                   000131          MICPC=MICPC+1
11146  027624  000533                   .WORD   .$.
11147  027626                           SBR     SCPE
```

```
11148            000132                          MICPC=MICPC+1
11149  027626    104454                          .WORD   .$.
11150  027630                                    SBR     1$              ;DO THE NEXT ITERATION...
11151            000133                          MICPC=MICPC+1
11152  027630    100400                          .WORD   .$.
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                                             ;  TEST 52
11169                                             ;--------------
11170  027632  012737  000052  001202    TST52:  MOV     #52,STSTNM              ; LOAD THE NO. OF THIS TEST
11171  027640  012737  030160  001442            MOV     #TST53,NEXT             ; POINT TO THE START OF NEXT TEST.
11172                                                                    ;R1 CONTAINS BASE KMC11 ADDRESS
11173  027646  004737  035536                    JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
11174  027652  027666                            MCT52
11175  027654  104022                            ERROR   22                      ;TIME OUT ERROR...
11176  027656  012706  001200                    MOV     #STACK,SP               ;RESET STACK...
11177  027662  000177  151554                    JMP     @NEXT                   ;GO TO NEXT TEST...
11178  027666                            MCT52:
11179  027666                            1$:     MOVE    # 0,MLR          ;SET MAR+LO.
11180            000000                          MICPC=MICPC+1
11181  027666    010000                          .WORD   .$.
11182  027670                                    MOVE    # 0,MPR          ;SET MAR+HI.
11183            000001                          MICPC=MICPC+1
11184  027670    004000                          .WORD   .$.
11185  027672                                    MOVE    # 0,BREG                 ;
11186            000002                          MICPC=MICPC+1
11187  027672    000400                          .WORD   .$.
11188  027674                                    MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS...
11189            000003                          MICPC=MICPC+1
11190  027674    063236                          .WORD   .$.
11191  027676                                    MOVE    BREG,SPAD <0>   ;
11192            000004                          MICPC=MICPC+1
11193  027676    063220                          .WORD   .$.
11194  027700                                    MOVE    BREG,SPAD <1>   ;
11195            000005                          MICPC=MICPC+1
11196  027700    063221                          .WORD   .$.
11197  027702                                    MOVE    BREG,SPAD <2>   ;
11198            000006                          MICPC=MICPC+1
11199  027702    063222                          .WORD   .$.
11200  027704                                    $DEC    SPAD <2>                ;
11201            000007                          MICPC=MICPC+1
11202  027704    063162                  .WORD   .$.!.DSP
11203  027706                                    MOVE    BREG,SPAD <4>           ;
```

```
11204                   000010
11205  027706  063224           MICPC=MICPC+1
11206  027710                   .WORD   .$.
11207                   000011   MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
11208  027710  016400           MICPC=MICPC+1
11209  027712                   .WORD   .$.
11210                   000012   MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
11211  027712  016400           MICPC=MICPC+1
11212  027714                   .WORD   .$.
11213                   000013   MOVE    # 0,MEM MARINC
11214  027714  016400           MICPC=MICPC+1
11215  027716                   .WORD   .$.
11216                   000014   MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
11217  027716  016400           MICPC=MICPC+1
11218  027720                   .WORD   .$.
11219                   000015   MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
11220  027720  016777           MICPC=MICPC+1
11221  027722                   .WORD   .$.
11222                   000016   MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
11223  027722  016400           MICPC=MICPC+1
11224  027724                   .WORD   .$.
11225                   000017   MOVE    # -1,MEM MARINC
11226  027724  016777           MICPC=MICPC+1
11227  027726                   .WORD   .$.
11228                   000020   MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
11229  027726  016777           MICPC=MICPC+1
.123[  027730                   .WORD   .$.
:[31                    000021   MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
11232  027730  016400           MICPC=MICPC+1
11233  027732                   .WORD   .$.
11234                   000022   MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
11235  027732  016777           MICPC=MICPC+1
11236  027734                   .WORD   .$.
11237                   000023   MOVE    # 1,MEM MARINC
11238  027734  016401           MICPC=MICPC+1
11239  027736                   .WORD   .$.
11240                   000024   MOVE    # 1,MEM MARINC  ;RESULT WITH C BIT SET.
11241  027736  016401           MICPC=MICPC+1
11242  027740                   .WORD   .$.
11243                   000025   MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
11244  027740  016777           MICPC=MICPC+1
11245  027742                   .WORD   .$.
11246                   000026   MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
11247  027742  016777           MICPC=MICPC+1
11248  027744                   .WORD   .$.
11249                   000027   MOVE    # 0,MEM MARINC
11250  027744  016400           MICPC=MICPC+1
11251  027746                   .WORD   .$.
11252                   000030   MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
11253  027746  016400           MICPC=MICPC+1
11254  027750                   .WORD   .$.
11255                   000031   MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11256  027750  016525           MICPC=MICPC+1
11257  027752                   .WORD   .$.
11258                   000032   MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11259  027752  016525           MICPC=MICPC+1
                                .WORD   .$.
```

```
11260  027754
11261         000033
11262  027754 016400
11263  027756
11264         000034
11265  027756 016400
11266  027760
11267         000035
11268  027760 016652
11269  027762
11270         000036
11271  027762 016525
11272  027764
11273         000037
11274  027764 016525
11275  027766
11276         000040
11277  027766 016525
11278  027770
11279         000041
11280  027770 016525
11281  027772
11282         000042
11283  027772 016652
11284  027774
11285         000043
11286  027774 016653
11287  027776
11288         000044
11289  027776 016653
11290  030000
11291         000045
11292  030000 016652
11293  030002
11294         000046
11295  030002 016652
11296  030004
11297         000047
11298  030004 016400
11299  030006
11300         000050
11301  030006 016400
11302  030010
11303         000051
11304  030010 003007
11305
11306  030012
11307         000052
11308  030012 010000
11309
11310  030014
11311         000053
11312  030014 000400
11313  030016
11314         000054
11315  030016 060401
```

```
        MOVE    # 0,MEM MARINC
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 125,MEM MARINC
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 125,MEM MARINC        ;RESULT WITH C BIT SET.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 253,MEM MARINC
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 253,MEM MARINC        ;RESULT WITH C BIT SET.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 7,SPAD <7>    ;SET THE COUNT.
        MICPC=MICPC+1
        .WORD   .$.

        MOVE    # 0,MLR         ;MAR←0.
        MICPC=MICPC+1
        .WORD   .$.

25:     MOVE    # 0,BREG                ;
        MICPC=MICPC+1
        .WORD   .$.
        $ADD    SPAD <1>,BREG  ;CLEAR C BIT.
        MICPC=MICPC+1
        .WORD   .$.!.DD
```

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

```
11372  030062                          SBR     2$                      ; SCOPE THE DATA....
                 000076                 MICPC=MICPC+1
11374  030062    100453                .WORD   .$.
11375  030064                  6$:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
                 000077                 MICPC=MICPC+1
11377  030064    070204                .WORD   .$.
11378  030066                          MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
                 000100                 MICPC=MICPC+1
11380  030066    057220                .WORD   .$.
11381  030070                          MOVE    # 377,BREG      ;
                 000101                 MICPC=MICPC+1
11383  030070    000777                .WORD   .$.
11384  030072                          $GOOD   SPAD <2>,BREG   ;SET C BIT...
                 000102                 MICPC=MICPC+1
11386  030072    060402                .WORD   .$.!.DD
11387  030074                          SUB1C   MEM,SPAD <0>,BR.SP,MARINC       ;SP 0 & BR = SUB
                 000103                 MICPC=MICPC+1
11389  030074    057740                .WORD   .$.
11390  030076                          MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
                 000104                 MICPC=MICPC+1
11392  030076    057223                .WORD   .$.
11393  030100                          $IFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...
11394
11395
11396  030100                          SUB2C   SPAD <0>,MEM,NOP
                 000105                 MICPC=MICPC+1
11398  030100    040360                .WORD   .$.
11399  030102                          BZ      9$
                 000106                 MICPC=MICPC+1
11401  030102    101521                .WORD   .$.
11402  030104                          MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
                 000107                 MICPC=MICPC+1
11404  030104    041224                .WORD   .$.
11405  030106                          MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
                 000110                 MICPC=MICPC+1
11407  030106    061225                .WORD   .$.
11408  030110                          MOVE    # 23,BREG               ;ERROR TYPE...
                 000111                 MICPC=MICPC+1
11410  030110    000423                .WORD   .$.
11411  030112                          MOVE    BREG,OUT1 <CSR3>        ;
                 000112                 MICPC=MICPC+1
11413  030112    061223                .WORD   .$.
11414  030114                          MOVE    # 16,BREG               ;ALU FUNCTION CODE.
                 000113                 MICPC=MICPC+1
11416  030114    000416                .WORD   .$.
11417  030116                          MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
                 000114                 MICPC=MICPC+1
11419  030116    061227                .WORD   .$.
11420  030120                          CALL    ERROR   ;REPORT ERROR...
11421  030120                          MOVE    # <MICPC+3>,BREG
                 000115                 MICPC=MICPC+1
11423  030120    000517                .WORD   .$.
11424  030122                          $BR     ERROR
                 000116                 MICPC=MICPC+1
11426  030122    104400                .WORD   .$.
11427  030124                          MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER.
```

```
11428                   000117                          MICPC=MICPC+1
11429  030124  070204                                   .WORD   .$.
11430  030126                                           $BR     6$                      ;LOOP ON ERROR...
11431                   000120                          MICPC=MICPC+1
11432  030126  100477                                   .WORD   .$.
11433  030130                                 9$:       CALL    $CP1                    ;SCOPE THE ERROR...
11434  030130                                           MOVE    # <MICPC+3>,BREG
11435                   000121                          MICPC=MICPC+1
11436  030130  000523                                   .WORD   .$.
11437  030132                                           $BR     $CP1
11438                   000122                          MICPC=MICPC+1
11439  030132  104427                                   .WORD   .$.
11440  030134                                           MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER...
11441                   000123                          MICPC=MICPC+1
11442  030134  070204                                   .WORD   .$.
11443  030136                                           $BR     6$                      ;SCOPE THE DATA...
11444                   000124                          MICPC=MICPC+1
11445  030136  100477                                   .WORD   .$.
11446  030140                                           MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
11447                   000125                          MICPC=MICPC+1
11448  030140  000404                                   .WORD   .$.
11449  030142                                           $ADD    BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
11450  030142  077004                                   .WORD   .$.!MARINC!.DO
11451  030144                                           $DEC    SPAD <7>                ;IS IT DONE??
11452                   000126                          MICPC=MICPC+1
11453  030144  063167                         .WORD     .$.!.DSP
11454  030146                                           BZ      4$                      ;YES, SCOPE THE TEST.
11455                   000127                          MICPC=MICPC+1
11456  030146  101532                                   .WORD   .$.
11457  030150                                           $BR     2$                      ;DO, THE NEXT.
11458                   000130                          MICPC=MICPC+1
11459  030150  100453                                   .WORD   .$.
11460  030152                                 4$:       CALL    $CPE                    ;SCOPE THE TEST...
11461  030152                                           MOVE    # <MICPC+3>,BREG
11462                   000131                          MICPC=MICPC+1
11463  030152  000533                                   .WORD   .$.
11464  030154                                           $BR     $CPE
11465                   000132                          MICPC=MICPC+1
11466  030154  104454                                   .WORD   .$.
11467  030156                                           $BR     1$                      ;DO THE NEXT ITERATION...
11468                   000133                          MICPC=MICPC+1
11469  030156  100400                                   .WORD   .$.
11470  030160                         $ALUT1    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                         $XZ
11472
11473
11474                                            ;************************** TEST 53 **************************
11475                                            ;*ALU TEST
11476                                            ;*TEST OF ALU FUNCTION ADD W/C WITH C BIT CLEARED.
11477                                            ;*TEST OF ALU FUNCTION ADD W/C WITH C BIT SET.
11478                                            ;*ALU FUNCTION (A PLUS B PLUS C)
11479                                            ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
11480                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
11481  030160                         $XZ
11482                                            ;;**************************************************************
11483
```

```
11484  030160                          $TSTN
11485
11486                                          ;  TEST 53
11487  030160  012737  000053  001202  TST53:  MOV     #53,$TSTNM              ; LOAD THE NO. OF THIS TEST
11488  030166  012737  030506  001442          MOV     #TST54,NEXT             ; POINT TO THE START OF NEXT TEST.
11489                                                                 ;R1 CONTAINS BASE KMC11 ADDRESS
11490  030174  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
11491  030200  030214                          MCTS3
11492  030202  104022                          ERROR   22                      ;TIME OUT ERROR...
11493  030204  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
11494  030210  000177  151226                  JMP     @NEXT                   ;GO TO NEXT TEST...
11495  030214                          MCTS3:
11496  030214                          1$:     MOVE    # 0,MLR         ;SET MAR+LO.
11497                  000000                  MICPC=MICPC+1
11498  030214  010000                          .WORD   .$.
11499  030216                          MOVE    # 0,MPR         ;SET MAR+HI.
11500                  000001                  MICPC=MICPC+1
11501  030216  004000                          .WORD   .$.
11502  030220                          MOVE    # 0,BREG                ;
11503                  000002                  MICPC=MICPC+1
11504  030220  000400                          .WORD   .$.
11505  030222                          MOVE    BREG,SPAD (16)          ;FOR RETURN ADDRESS...
11506                  000003                  MICPC=MICPC+1
11507  030222  063236                          .WORD   .$.
11508  030224                          MOVE    BREG,SPAD (0)   ;
11509                  000004                  MICPC=MICPC+1
11510  030224  063220                          .WORD   .$.
11511  030226                          MOVE    BREG,SPAD (1)   ;
11512                  000005                  MICPC=MICPC+1
11513  030226  063221                          .WORD   .$.
11514  030230                          MOVE    BREG,SPAD (2)   ;
11515                  000006                  MICPC=MICPC+1
11516  030230  063222                          .WORD   .$.
11517  030232                          SDEC    SPAD (2)                ;
11518                  000007                  MICPC=MICPC+1
11519  030232  063162          .WORD   .$.!.DSP
11520  030234                          MOVE    BREG,SPAD (4)           ;
11521                  000010                  MICPC=MICPC+1
11522  030234  063224                          .WORD   .$.
11523  030236                          MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
11524                  000011                  MICPC=MICPC+1
11525  030236  016400                          .WORD   .$.
11526  030240                          MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
11527                  000012                  MICPC=MICPC+1
11528  030240  016400                          .WORD   .$.
11529  030242                          MOVE    # 0,MEM MARINC
11530                  000013                  MICPC=MICPC+1
11531  030242  016400                          .WORD   .$.
11532  030244                          MOVE    # 1,MEM MARINC  ;RESULT WITH C BIT SET.
11533                  000014                  MICPC=MICPC+1
11534  030244  016401                          .WORD   .$.
11535  030246                          MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
11536                  000015                  MICPC=MICPC+1
11537  030246  016777                          .WORD   .$.
11538  030250                          MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
11539                  000016                  MICPC=MICPC+1
```

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

```
11596                  000041
11597  030316  016525
11598  030320
11599                  000042
11600  030320  016652
11601  030322
11602                  000043
11603  030322  016777
11604  030324
11605                  000044
11606  030324  016400
11607  030326
11608                  000045
11609  030326  016652
11610  030330
11611                  000046
11612  030330  016652
11613  030332
11614                  000047
11615  030332  016524
11616  030334
11617                  000050
11618  030334  016525
11619  030336
11620                  000051
11621  030336  003007
11622
11623  030340
11624                  000052
11625  030340  010000
11626
11627  030342
11628                  000053
11629  030342  000400
11630  030344
11631                  000054
11632  030344  060401
11633  030346
11634                  000055
11635  030346  057220
11636  030350
11637                  000056
11638  030350  057420
11639  030352
11640
11641
11642  030352
11643                  000057
11644  030352  040360
11645  030354
11646                  000060
11647  030354  101473
11648  030356
11649                  000061
11650  030356  041224
11651  030360
```

```
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # -1,MEM MARINC
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 0,MEM MARINC   ;RESULT WITH C BIT SET.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 124,MEM MARINC         ;LOAD THE DATA IN MEMORY.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 125,MEM MARINC         ;RESULT WITH C BIT SET.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 7,SPAD <7>     ;SET THE COUNT.
        MICPC=MICPC+1
        .WORD   .$.

        MOVE    # 0,MLR          ;MAR+0.
        MICPC=MICPC+1
        .WORD   .$.

2$:     MOVE    # 0,BREG                  ;
        MICPC=MICPC+1
        .WORD   .$.
        SADD    SPAD <1>,BREG   ;CLEAR C BIT.
        MICPC=MICPC+1
        .WORD   .$.!.DD
        MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
        MICPC=MICPC+1
        .WORD   .$.
        ADD4C   MEM,SPAD <0>,BR.SP,MARINC       ;SPAD  <0>:=SFUNC.
        MICPC=MICPC+1
        .WORD   .$.
        SIFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.


        SUB2C   SPAD <0>,MEM,NOP
        MICPC=MICPC+1
        .WORD   .$.
        BZ      3$
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,OUT1 <4>    ;GOOD DATA
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 <5>   ;BAD DATA.
```

```
11652                 000062
11653  030360  061225
11654  030362
11655                 000063
11656  030362  000415
11657  030364
11658                 000064
11659  030364  061223
11660  030366
11661                 000065
11662  030366  000401
11663  030370
11664                 000066
11665  030370  061227
11666  030372
11667  030372
11668                 000067
11669  030372  000471
11670  030374
11671                 000070
11672  030374  104400
11673  030376
11674                 000071
11675  030376  070204
11676  030400
11677                 000072
11678  030400  100453
11679  030402
11680  030402
11681                 000073
11682  030402  000475
11683  030404
11684                 000074
11685  030404  104427
11686  030406
11687                 000075
11688  030406  070204
11689  030410
11690                 000076
11691  030410  100453
11692  030412
11693                 000077
11694  030412  070204
11695  030414
11696                 000100
11697  030414  057220
11698  030416
11699                 000101
11700  030416  000777
11701  030420
11702                 000102
11703  030420  060402
11704  030422
11705                 000103
11706  030422  057420
11707  030424
```

```
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 15,BREG       ;SET TYPE OF ERROR.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 01,BREG       ;LOAD FUNCTION CODE...
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
                MICPC=MICPC+1
                .WORD   .$.
                CALL    EROR            ;ALU ADD W/C ERROR...
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     EROR
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     2$              ;LOOP ON ERROR...
                MICPC=MICPC+1
                .WORD   .$.
35:             CALL    SCP1
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     SCP1
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,MLR    ;
                MICPC=MICPC+1
                .WORD   .$.
                SBR     2$              ; SCOPE THE DATA....
                MICPC=MICPC+1
                .WORD   .$.
65:             MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 377,BREG      ;
                MICPC=MICPC+1
                .WORD   .$.
                SADD    SPAD <2>,BREG   ;SET C BIT...
                MICPC=MICPC+1
                .WORD   .$.!.DD
                ADDWC   MEM,SPAD <0>,BR.SP,MARINC       ;SP 0 & BR = ADD W/C
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
```

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

```
11764                 000125
11765  030466  000404                  MICPC=MICPC+1
11766  030470                          .WORD   .$.
11767  030470  077004                  $ADD    BREG,SPAD (4),MARINC    ;ALSO DATA POINTER.
11768  030472                          .WORD   .$.!MARINC!.DO
11769                 000126           $DEC    SPAD (7)                ;IS IT DONE??
11770  030472  063167                  MICPC=MICPC+1
.WORD                                   .$.!.DSP
11771  030474                          BZ      4$                      ;YES, SCOPE THE TEST.
11772                 000127           MICPC=MICPC+1
11773  030474  101532                  .WORD   .$.
11774  030476                          $BR     2$                      ;DO, THE NEXT.
11775                 000130           MICPC=MICPC+1
11776  030476  100453                  .WORD   .$.
11777  030500                  4$:     CALL    SCPE                    ;SCOPE THE TEST...
11778  030500                          MOVE    # (MICPC+3),BREG
11779                 000131           MICPC=MICPC+1
11780  030500  000533                  .WORD   .$.
11781  030502                          $BR     SCPE
11782                 000132           MICPC=MICPC+1
11783  030502  104454                  .WORD   .$.
11784  030504                          $BR     1$                      ;DO THE NEXT ITERATION...
11785                 000133           MICPC=MICPC+1
11786  030504  100400                  .WORD   .$.
11787  030506                  $ALUT1  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                                   ;;***************************************************************
11800
11801  030506                  $TSTN
11802
11803                                   ;  TEST 54
                                        ;-------------
11804  030506  012737  000054  001202  TST54:  MOV     #54,$TSTNM              ; 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,LDVRWT               ;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                  1$:     MOVE    # 0,MLR          ;SET MAR<-LO.
11814                 000000           MICPC=MICPC+1
11815  030542  010000                  .WORD   .$.
11816  030544                          MOVE    # 0,MPR          ;SET MAR<-HI.
11817                 000001           MICPC=MICPC+1
11818  030544  004000                  .WORD   .$.
11819  030546                          MOVE    # 0,BREG                 ;
```

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

```
11875  030614                  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
11877          000025          MICPC=MICPC+1
11878  030614  016777          .WORD   .$.
11879  030616                  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
11880          000026          MICPC=MICPC+1
11881  030616  016777          .WORD   .$.
11882  030620                  MOVE    # -1,MEM MARINC
11883          000027          MICPC=MICPC+1
11884  030620  016777          .WORD   .$.
11885  030622                  MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
11886          000030          MICPC=MICPC+1
11887  030622  016400          .WORD   .$.
11888  030624                  MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
11889          000031          MICPC=MICPC+1
11890  030624  016525          .WORD   .$.
11891  030626                  MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
11892          000032          MICPC=MICPC+1
11893  030626  016525          .WORD   .$.
11894  030630                  MOVE    # -1,MEM MARINC
11895          000033          MICPC=MICPC+1
11896  030630  016777          .WORD   .$.
11897  030632                  MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
11898          000034          MICPC=MICPC+1
11899  030632  016400          .WORD   .$.
11900  030634                  MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
11901          000035          MICPC=MICPC+1
11902  030634  016652          .WORD   .$.
11903  030636                  MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
11904          000036          MICPC=MICPC+1
11905  030636  016525          .WORD   .$.
11906  030640                  MOVE    # 124,MEM MARINC
11907          000037          MICPC=MICPC+1
11908  030640  016524          .WORD   .$.
11909  030642                  MOVE    # 125,MEM MARINC         ;RESULT WITH C BIT SET.
11910          000040          MICPC=MICPC+1
11911  030642  016525          .WORD   .$.
11912  030644                  MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
11913          000041          MICPC=MICPC+1
11914  030644  016525          .WORD   .$.
11915  030646                  MOVE    # 252,MEM MARINC         ·LOAD THE DATA IN MEMORY
11916          000042          MICPC=MICPC+1
11917  030646  016652          .WORD   .$.
11918  030650                  MOVE    # 252,MEM MARINC
11919          000043          MICPC=MICPC+1
11920  030650  016652          .WORD   .$.
11921  030652                  MOVE    # 253,MEM MARINC         ;RESULT WITH C BIT SET.
11922          000044          MICPC=MICPC+1
11923  030652  016653          .WORD   .$.
11924  030654                  MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
11925          000045          MICPC=MICPC+1
11926  030654  016652          .WORD   .$.
11927  030656                  MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
11928          000046          MICPC=MICPC+1
11929  030656  016652          .WORD   .$.
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                          MICPC=MICPC+1
11941  030662  000050          .WORD   .S.
11942  030664  016400          MOVE    # 7,SPAD <7>    ;SET THE COUNT.
11943  030664                  MICPC=MICPC+1
11944                  000051  .WORD   .S.
11945  030664  003007
11946
11947  030666                  MOVE    # 0,MLR         ;MAR←0.
11948                          MICPC=MICPC+1
11949                  000052  .WORD   .S.
11950  030666  010000
11951
11952  030670          2$:     MOVE    # 0,BREG                 ;
11953                          MICPC=MICPC+1
11954                  000053  .WORD   .S.
11955  030670  000400          SADD    SPAD <1>,BREG   ;CLEAR C BIT.
11956  030672                  MICPC=MICPC+1
11957                  000054  .WORD   .S.!.DO
11958  030672  060401          MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
11959  030674                  MICPC=MICPC+1
11960                  000055  .WORD   .S.
11961  030674  057220          SUBWC   MEM,SPAD <0>,BR.SP,MARINC       ;SPAD  <0>:=$FUNC.
11962  030676                  MICPC=MICPC+1
11963                  000056  .WORD   .S.
11964  030676  057440          $IFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
11965  030700
11966
11967  030700          3$:     SUB2C   SPAD <0>,MEM,NOP
11968                          MICPC=MICPC+1
11969                  000057  .WORD   .S.
11970  030700  040360          BZ      33
11971  030702                  MICPC=MICPC+1
11972                  000060  .WORD   .S.
11973  030702  101473          MOVE    MEM,OUT1 <4>    ;GOOD DATA
11974  030704                  MICPC=MICPC+1
11975                  000061  .WORD   .S.
11976  030704  041224          MOVE    BREG,OUT1 <5>   ;BAD DATA.
11977  030706                  MICPC=MICPC+1
11978                  000062  .WORD   .S.
11979  030706  061225          MOVE    # 15,BREG       ;SET TYPE OF ERROR.
11980  030710                  MICPC=MICPC+1
11981                  000063  .WORD   .S.
11982  030710  000415          MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
11983  030712                  MICPC=MICPC+1
11984                  000064  .WORD   .S.
11985  030712  061223          MOVE    # 02,BREG       ;LOAD FUNCTION CODE...
11986  030714                  MICPC=MICPC+1
11987                  000065  .WORD   .S.
11988  030714  000402          MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
11989  030716                  MICPC=MICPC+1
11990                  000066  .WORD   .S.
11991  030716  061227          CALL    EROR            ;ALU SUB W/C ERROR...
11992  030720                  MOVE    # <MICPC+3>,BREG
11993  030720                  MICPC=MICPC+1
11994                  000067  .WORD   .S.
11995  030720  000471          $BR     EROR
11996  030722
```

```
11988                   000070
11989   030722  104400
11990   030724
11991                   000071
11992   030724  070204
11993   030726
11994                   000072
11995   030726  100453
11996   030730
11997   030730
11998                   000073
11999   030730  000475
12000   030732
12001                   000074
12002   030732  104427
12003   030734
12004                   000075
12005   030734  070204
12006   030736
12007                   000076
12008   030736  100453
12009   030740
12010                   000077
12011   030740  070204
12012   030742
12013                   000100
12014   030742  057220
12015   030744
12016                   000101
12017   030744  000777
12018   030746
12019                   000102
12020   030746  060402
12021   030750
12022                   000103
12023   030750  057440
12024   030752
12025                   000104
12026   030752  057223
12027   030754
12028
12029
12030   030754
12031                   000105
12032   030754  040360
12033   030756
12034                   000106
12035   030756  101521
12036   030760
12037                   000107
12038   030760  041224
12039   030762
12040                   000110
12041   030762  061225
12042   030764
12043                   000111
```

```
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
        MICPC=MICPC+1
        .WORD   .$.
        SBR     2$              ;LOOP ON ERROR...
        MICPC=MICPC+1
        .WORD   .$.
3$:     CALL    SCP1
        MOVE    # <MICPC+3>,BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     SCP1
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD <4>,MLR    ;
        MICPC=MICPC+1
        .WORD   .$.
        SBR     2$              ; SCOPE THE DATA....
        MICPC=MICPC+1
        .WORD   .$.
6$:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 377,BREG      ;
        MICPC=MICPC+1
        .WORD   .$.
        $GOOD   SPAD <2>,BREG   ;SET C BIT...
        MICPC=MICPC+1
        .WORD   .$.!.DD
        SUBWC   MEM,SPAD <0>,BR.SP,MARINC       ;SP 0 & BR = SUB W/C
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
        MICPC=MICPC+1
        .WORD   .$.
        $IFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...


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

```
12044  030764  000423                .WORD   .$.
12045  030766                        MOVE    BREG,OUT1 (CSR3)        ;
12046          000112                MICPC=MICPC+1
12047  030766  061223                .WORD   .$.
12048  030770                        MOVE    # D2,BREG               ;ALU FUNCTION CODE.
12049          000113                MICPC=MICPC+1
12050  030770  000402                .WORD   .$.
12051  030772                        MOVE    BREG,OUT1 (CSR7)        ;LOAD IT...
12052          000114                MICPC=MICPC+1
12053  030772  061227                .WORD   .$.
12054  030774                        CALL    ERROR   ;REPORT ERROR...
12055  030774                        MOVE    # (MICPC+3),BREG
12056          000115                MICPC=MICPC+1
12057  030774  000517                .WORD   .$.
12058  030776                        SBR     ERROR
12059          000116                MICPC=MICPC+1
12060  030776  104400                .WORD   .$.
12061  031000                        MOVE    SPAD (4),MLR            ;RESTORE DATA POINTER.
12062          000117                MICPC=MICPC+1
12063  031000  070204                .WORD   .$.
12064  031002                        SBR     6$                      ;LOOP ON ERROR...
12065          000120                MICPC=MICPC+1
12066  031002  100477                .WORD   .$.
12067  031004                9$:     CALL    SCP1            ;SCOPE THE ERROR...
12068  031004                        MOVE    # (MICPC+3),BREG
12069          000121                MICPC=MICPC+1
12070  031004  000523                .WORD   .$.
12071  031006                        SBR     SCP1
12072          000122                MICPC=MICPC+1
12073  031006  104427                .WORD   .$.
12074  031010                        MOVE    SPAD (4),MLR    ;RESTORE DATA POINTER...
12075          000123                MICPC=MICPC+1
12076  031010  070204                .WORD   .$.
12077  031012                        SBR     6$              ;SCOPE THE DATA...
12078          000124                MICPC=MICPC+1
12079  031012  100477                .WORD   .$.
12080  031014                        MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
12081          000125                MICPC=MICPC+1
12082  031014  000404                .WORD   .$.
12083  031016                        SADD    BREG,SPAD (4),MARINC    ;ALSO DATA POINTER.
12084  031016  077004                .WORD   .$.!MARINC!.DD
12085  031020                        SDEC    SPAD (7)                ;IS IT DONE??
12086          000126                MICPC=MICPC+1
12087  031020  063167                .WORD   .$.!.DSP
12088  031022                        BZ      4$              ;YES, SCOPE THE TEST.
12089          000127                MICPC=MICPC+1
12090  031022  101532                .WORD   .$.
12091  031024                        SBR     2$              ;DO, THE NEXT.
12092          000130                MICPC=MICPC+1
12093  031024  100453                .WORD   .$.
12094  031026                4$:     CALL    SCPE            ;SCOPE THE TEST...
12095  031026                        MOVE    # (MICPC+3),BREG
12096          000131                MICPC=MICPC+1
12097  031026  000533                .WORD   .$.
12098  031030                        SBR     SCPE
12099          000132                MICPC=MICPC+1
```

```
12100  031030  104454                          .WORD   .$.
12101  031032                                  SBR     1$              ;DO THE NEXT ITERATION...
12102          000133                          MICPC=MICPC+1
12103  031034  100400                          .WORD   .$.
12104  031034                          SALUT1  0,<INC A>,$INC,1,1,0,0,1,1,0,0,126,126,253,253,126,126,253,253,<A PLUS 1>,0,03
12105  031034                          $XZ
12106
12107
12108                                           ;************************** TEST 55 **************************
12109                                           ;*ALU TEST
12110                                           ;*TEST OF ALU FUNCTION INC A WITH C BIT CLEARED.
12111                                           ;*TEST OF ALU FUNCTION INC A WITH C BIT SET.
12112                                           ;*ALU FUNCTION (A PLUS 1)
12113                                           ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
12114                                           ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
12115  031034                          $XZ
12116                                           ;:**********************************************************************
12117
12118  031034                          $TSTN
12119                                           ;  TEST 55
12120                                           ;-------------
12121  031034  012737  000055  001202  TST55:  MOV     #55,$TSTNM              ; LOAD THE NO. OF THIS TEST
12122  031042  012737  031362  001442          MOV     #TST56,NEXT             ; POINT TO THE START OF NEXT TEST.
12123                                                                  ;R1 CONTAINS BASE KMC11 ADDRESS
12124  031050  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
12125  031054  031070                          MCT55
12126  031056  104022                          ERROR   22                      ;TIME OUT ERROR...
12127  031060  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
12128  031064  000177  150352                  JMP     @NEXT                   ;GO TO NEXT TEST...
12129  031070                          MCT55:
12130  031070                          1$:     MOVE    # 0,MLR          ;SET MAR+LO.
12131          000000                          MICPC=MICPC+1
12132  031070  010000                          .WORD   .$.
12133  031072                                  MOVE    # 0,MPR          ;SET MAR+HI.
12134          000001                          MICPC=MICPC+1
12135  031072  004000                          .WORD   .$.
12136  031074                                  MOVE    # 0,BREG                 ;
12137          000002                          MICPC=MICPC+1
12138  031074  000400                          .WORD   .$.
12139  031076                                  MOVE    BREG,SPAD (16)          ;FOR RETURN ADDRESS...
12140          000003                          MICPC=MICPC+1
12141  031076  063236                          .WORD   .$.
12142  031100                                  MOVE    BREG,SPAD (0)   ;
12143          000004                          MICPC=MICPC+1
12144  031100  063220                          .WORD   .$.
12145  031102                                  MOVE    BREG,SPAD (1)   ;
12146          000005                          MICPC=MICPC+1
12147  031102  063221                          .WORD   .$.
12148  031104                                  MOVE    BREG,SPAD (2)   ;
12149          000006                          MICPC=MICPC+1
12150  031104  063222                          .WORD   .$.
12151  031106                                  SDEC    SPAD (2)                ;
12152          000007                          MICPC=MICPC+1
12153  031106  063162                  .WORD   .$.!.DSP
12154  031110                                  MOVE    BREG,SPAD (4)           ;
12155          000010                          MICPC=MICPC+1
```

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

```
12212                   000033
12213  031156  016526
12214  031160
12215                   000034
12216  031160  016526
12217  031162
12218                   000035
12219  031162  016652
12220  031164
12221                   000036
12222  031164  016525
12223  031166
12224                   000037
12225  031166  016653
12226  031170
12227                   000040
12228  031170  016653
12229  031172
12230                   000041
12231  031172  016525
12232  031174
12233                   000042
12234  031174  016652
12235  031176
12236                   000043
12237  031176  016526
12238  031200
12239                   000044
12240  031200  016526
12241  031202
12242                   000045
12243  031202  016652
12244  031204
12245                   000046
12246  031204  016652
12247  031206
12248                   000047
12249  031206  016653
12250  031210
12251                   000050
12252  031210  016653
12253  031212
12254                   000051
12255  031212  003007
12256
12257  031214
12258                   000052
12259  031214  010000
12260
12261  031216
12262                   000053
12263  031216  000400
12264  031220
12265                   000054
12266  031220  060401
12267  031222
```

```
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    # 126,MEM MARINC         ;RESULT WITH C BIT SET.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    # 253,MEM MARINC
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    # 253,MEM MARINC         ;RESULT WITH C BIT SET.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    # 126,MEM MARINC
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    # 126,MEM MARINC         ;RESULT WITH C BIT SET.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    # 253,MEM MARINC         ;LOAD THE DATA IN MEMORY.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    # 253,MEM MARINC         ;RESULT WITH C BIT SET.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    # 7,SPAD <7>     ;SET THE COUNT.
                        MICPC=MICPC+1
                        .WORD   .$.

                        MOVE    # 0,MLR          ;MAR+0.
                        MICPC=MICPC+1
                        .WORD   .$.

25:                     MOVE    # 0,BREG                 ;
                        MICPC=MICPC+1
                        .WORD   .$.
                        SADD    SPAD <1>,BREG   ;CLEAR C BIT.
                        MICPC=MICPC+1
                        .WORD   .$.!.DD
                        MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
```

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

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

```
12380  031326  070204                .WORD   .$.
12381  031330                        $BR     6$                      ;LOOP ON ERROR...
12382          000120                MICPC=MICPC+1
12383  031330  100477                .WORD   .$.
12384  031332                9$:     CALL    $CP1                    ;SCOPE THE ERROR...
12385  031332                        MOVE    # <MICPC+3>,BREG
12386          000121                MICPC=MICPC+1
12387  031332  000523                .WORD   .$.
12388  031334                        $BR     $CP1
12389          000122                MICPC=MICPC+1
12390  031334  104427                .WORD   .$.
12391  031336                        MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER...
12392          000123                MICPC=MICPC+1
12393  031336  070204                .WORD   .$.
12394  031340                        $BR     6$                      ;SCOPE THE DATA...
12395          000124                MICPC=MICPC+1
12396  031340  100477                .WORD   .$.
12397  031342                        MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
12398          000125                MICPC=MICPC+1
12399  031342  000404                .WORD   .$.
12400  031344                        $ADD    BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
12401  031344  077004                .WORD   .$.!MARINC!.DD
12402  031346                        $DEC    SPAD <7>                ;IS IT DONE??
12403          000126                MICPC=MICPC+1
12404  031346  063167        .WORD   .$.!.DSP
12405  031350                        BZ      4$                      ;YES, SCOPE THE TEST.
12406          000127                MICPC=MICPC+1
12407  031350  101532                .WORD   .$.
12408  031352                        $BR     2$                      ;DO, THE NEXT.
12409          000130                MICPC=MICPC+1
12410  031352  100453                .WORD   .$.
12411  031354                4$:     CALL    $CPE                    ;SCOPE THE TEST...
12412  031354                        MOVE    # <MICPC+3>,BREG
12413          000131                MICPC=MICPC+1
12414  031354  000533                .WORD   .$.
12415  031356                        $BR     $CPE
12416          000132                MICPC=MICPC+1
12417  031356  104454                .WORD   .$.
12418  031360                        $BR     1$                      ;DO THE NEXT ITERATION...
12419          000133                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                $TSTN
```

```
12436
12437                                                   ;  TEST 56
12438                                                   ;--------------
12439  031362  012737  000056  001202  TST56:  MOV     #56,$TSTNM              ; LOAD THE NO. OF THIS TEST
12440  031370  012737  031710  001442          MOV     #TST57,NEXT             ; POINT TO THE START OF NEXT TEST.
12441                                                                   ;R1 CONTAINS BASE KMC11 ADDRESS
12442  031376  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
12443  031402  031416                          MCT56
12444  031404  104022                          ERROR   22                      ;TIME OUT ERROR...
12445  031406  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
12446  031412  000177  150024                  JMP     @NEXT                   ;GO TO NEXT TEST...
12447  031416                          MCT56:
12448  031416                          1$:     MOVE    # 0,MLR          ;SET MAR←LO.
12449          000000                          MICPC=MICPC+1
12450  031416  010000                          .WORD   .$.
12451  031420                                  MOVE    # 0,MPR          ;SET MAR←HI.
12452          000001                          MICPC=MICPC+1
12453  031420  004000                          .WORD   .$.
12454  031422                                  MOVE    # 0,BREG                 ;
12455          000002                          MICPC=MICPC+1
12456  031422  000400                          .WORD   .$.
12457  031424                                  MOVE    BREG,SPAD (16)          ;FOR RETURN ADDRESS...
12458          000003                          MICPC=MICPC+1
12459  031424  063236                          .WORD   .$.
12460  031426                                  MOVE    BREG,SPAD (0)   ;
12461          000004                          MICPC=MICPC+1
12462  031426  063220                          .WORD   .$.
12463  031430                                  MOVE    BREG,SPAD (1)   ;
12464          000005                          MICPC=MICPC+1
12465  031430  063221                          .WORD   .$.
12466  031432                                  MOVE    BREG,SPAD (2)   ;
12467          000006                          MICPC=MICPC+1
12468  031432  063222                          .WORD   .$.
12469  031434                                  SDEC    SPAD (2)                ;
12470          000007                          MICPC=MICPC+1
12471  031434  063162                  .WORD   .$.!.DSP
12472  031436                                  MOVE    BREG,SPAD (4)           ;
12473          000010                          MICPC=MICPC+1
12474  031436  063224                          .WORD   .$.
12475  031440                                  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
12476          000011                          MICPC=MICPC+1
12477  031440  016400                          .WORD   .$.
12478  031442                                  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
12479          000012                          MICPC=MICPC+1
12480  031442  016400                          .WORD   .$.
12481  031444                                  MOVE    # 0,MEM MARINC
12482          000013                          MICPC=MICPC+1
12483  031444  016400                          .WORD   .$.
12484  031446                                  MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
12485          000014                          MICPC=MICPC+1
12486  031446  016400                          .WORD   .$.
12487  031450                                  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
12488          000015                          MICPC=MICPC+1
12489  031450  016777                          .WORD   .$.
12490  031452                                  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
12491          000016                          MICPC=MICPC+1
12492  031452  016400                          .WORD   .$.
```

```
12492 031454
12493         000017
12494 031454  016776
12495 031456
12496         000020
12497 031456  016776
12498 031460
12499         000021
12500 031460  016400
12501 031462
12502         000022
12503 031462  016777
12504 031464
12505         000023
12506 031464  016400
12507 031466
12508         000024
12509 031466  016400
12510 031470
12511         000025
12512 031470  016777
12513 031472
12514         000026
12515 031472  016777
12516 031474
12517         000027
12518 031474  016776
12519 031476
12520         000030
12521 031476  016776
12522 031500
12523         000031
12524 031500  016525
12525 031502
12526         000032
12527 031502  016525
12528 031504
12529         000033
12530 031504  016652
12531 031506
12532         000034
12533 031506  016652
12534 031510
12535         000035
12536 031510  016652
12537 031512
12538         000036
12539 031512  016525
12540 031514
12541         000037
12542 031514  016524
12543 031516
12544         000040
12545 031516  016524
12546 031520
12547         000041
```

```
MOVE    # 376,MEM MARINC
MICPC=MICPC+1
.WORD   .$.
MOVE    # 376,MEM MARINC         ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 0,MEM MARINC   ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .$.
MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 0,MEM MARINC
MICPC=MICPC+1
.WORD   .$.
MOVE    # 0,MEM MARINC   ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD   .$.
MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
MICPC=MICPC+1
.WORD   .$.
MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 376,MEM MARINC
MICPC=MICPC+1
.WORD   .$.
MOVE    # 376,MEM MARINC         ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 252,MEM MARINC
MICPC=MICPC+1
.WORD   .$.
MOVE    # 252,MEM MARINC         ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 124,MEM MARINC
MICPC=MICPC+1
.WORD   .$.
MOVE    # 124,MEM MARINC         ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD   .$.
MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
```

```
12548  031520  016525          .WORD   .$.
12549  031522                  MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY
12550          000042          MICPC=MICPC+1
12551  031522  016652          .WORD   .$.
12552  031524                  MOVE    # 252,MEM MARINC
12553          000043          MICPC=MICPC+1
12554  031524  016652          .WORD   .$.
12555  031526                  MOVE    # 252,MEM MARINC        ;RESULT WITH C BIT SET.
12556          000044          MICPC=MICPC+1
12557  031526  016652          .WORD   .$.
12558  031530                  MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12559          000045          MICPC=MICPC+1
12560  031530  016652          .WORD   .$.
12561  031532                  MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12562          000046          MICPC=MICPC+1
12563  031532  016652          .WORD   .$.
12564  031534                  MOVE    # 124,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12565          000047          MICPC=MICPC+1
12566  031534  016524          .WORD   .$.
12567  031536                  MOVE    # 124,MEM MARINC        ;RESULT WITH C BIT SET.
12568          000050          MICPC=MICPC+1
12569  031536  016524          .WORD   .$.
12570  031540                  MOVE    # 7,SPAD (7)     ;SET THE COUNT.
12571          000051          MICPC=MICPC+1
12572  031540  003007          .WORD   .$.
12573
12574  031542                  MOVE    # 0,MLR          ;MAR+0.
12575          000052          MICPC=MICPC+1
12576  031542  010000          .WORD   .$.
12577
12578  031544          2$:     MOVE    # 0,BREG                  ;
12579          000053          MICPC=MICPC+1
12580  031544  000400          .WORD   .$.
12581  031546                  SADD    SPAD (1),BREG   ;CLEAR C BIT.
12582          000054          MICPC=MICPC+1
12583  031546  060401          .WORD   .$.!.DD
12584  031550                  MOVE    MEM,SPAD (0) MARINC     ;GET THE FIRST OPERAND.
12585          000055          MICPC=MICPC+1
12586  031550  057220          .WORD   .$.
12587  031552                  $ASL    SPAD (0),BR.SP,MARINC   ;
12588          000056          MICPC=MICPC+1
12589  031552  077520          .WORD   .$.
12590  031554                  $IFEQ   MEM,SPAD (0) 3$ ;BRANCH IF GOOD.
12591
12592
12593  031554                  SUB2C   SPAD (0),MEM,NOP
12594          000057          MICPC=MICPC+1
12595  031554  040360          .WORD   .$.
12596  031556                  BZ      3$
12597          000060          MICPC=MICPC+1
12598  031556  101473          .WORD   .$.
12599  031560                  MOVE    MEM,OUT1 (4)    ;GOOD DATA
12600          000061          MICPC=MICPC+1
12601  031560  041224          .WORD   .$.
12602  031562                  MOVE    BREG,OUT1 (5)   ;BAD DATA.
12603          000062          MICPC=MICPC+1
```

```
12604  031562  061225                .WORD   .$.
12605  031564                        MOVE    # 15,BREG       ;SET TYPE OF ERROR.
12606          000063                MICPC=MICPC+1
12607  031564  000415                .WORD   .$.
12608  031566                        MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
12609          000064                MICPC=MICPC+1
12610  031566  061223                .WORD   .$.
12611  031570                        MOVE    # 05,BREG       ;LOAD FUNCTION CODE...
12612          000065                MICPC=MICPC+1
12613  031570  000405                .WORD   .$.
12614  031572                        MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
12615          000066                MICPC=MICPC+1
12616  031572  061227                .WORD   .$.
12617  031574                        CALL    ERROR           ;ALU 2A ERROR...
12618  031574                        MOVE    # <MICPC+3>,BREG
12619          000067                MICPC=MICPC+1
12620  031574  000471                .WORD   .$.
12621  031576                        SBR     ERROR
12622          000070                MICPC=MICPC+1
12623  03157o  104400                .WORD   .$.
12624  031600                        MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
12625          000071                MICPC=MICPC+1
12626  031600  070204                .WORD   .$.
12627  031602                        SBR     2$              ;LOOP ON ERROR...
12628          000072                MICPC=MICPC+1
12629  031602  100453                .WORD   .$.
12630  031604                3$:     CALL    SCP1
12631  031604                        MOVE    # <MICPC+3>,BREG
12632          000073                MICPC=MICPC+1
12633  031604  000475                .WORD   .$.
12634  031606                        SBR     SCP1
12635          000074                MICPC=MICPC+1
12636  031606  104427                .WORD   .$.
12637  031610                        MOVE    SPAD <4>,MLR    ;
12638          000075                MICPC=MICPC+1
12639  031610  070204                .WORD   .$.
12640  031612                        SBR     2$              ; SCOPE THE DATA....
12641          000076                MICPC=MICPC+1
12642  031612  100453                .WORD   .$.
12643  031614                6$:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
12644          000077                MICPC=MICPC+1
12645  031614  070204                .WORD   .$.
12646  031616                        MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
12647          000100                MICPC=MICPC+1
12648  031616  057220                .WORD   .$.
12649  031620                        MOVE    # 377,BREG      ;
12650          000101                MICPC=MICPC+1
12651  031620  000777                .WORD   .$.
12652  031622                        SADD    SPAD <2>,BREG   ;SET C BIT...
12653          000102                MICPC=MICPC+1
12654  031622  060402                .WORD   .$.!.D(
12655  031624                        SASL    SPAD <0>,BR.SP,MARINC           ;SP 0 & BR = 2A
12656          000103                MICPC=MICPC+1
12657  031624  077520                .WORD   .$.
12658  031626                        MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
12659          000104                MICPC=MICPC+1
```

```
12660  031626  057223                  .WORD   .$.
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   .$.
12667  031632                          BZ      9$
12668          000106                  MICPC=MICPC+1
12669  031632  101521                  .WORD   .$.
12670  031634                          MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
12671          000107                  MICPC=MICPC+1
12672  031634  041224                  .WORD   .$.
12673  031636                          MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
12674          000110                  MICPC=MICPC+1
12675  031636  061225                  .WORD   .$.
12676  031640                          MOVE    # 23,BREG               ;ERROR TYPE...
12677          000111                  MICPC=MICPC+1
12678  031640  000423                  .WORD   .$.
12679  031642                          MOVE    BREG,OUT1 <CSR3>        ;
12680          000112                  MICPC=MICPC+1
12681  031642  061223                  .WORD   .$.
12682  031644                          MOVE    # 05,BREG               ;ALU FUNCTION CODE.
12683          000113                  MICPC=MICPC+1
12684  031644  000405                  .WORD   .$.
12685  031646                          MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
12686          000114                  MICPC=MICPC+1
12687  031646  061227                  .WORD   .$.
12688  031650                          CALL    ERROR   ;REPORT ERROR...
12689  031650                          MOVE    # <MICPC+3>,BREG
12690          000115                  MICPC=MICPC+1
12691  031650  000517                  .WORD   .$.
12692  031652                          SBR     ERROR
12693          000116                  MICPC=MICPC+1
12694  031652  104400                  .WORD   .$.
12695  031654                          MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER.
12696          000117                  MICPC=MICPC+1
12697  031654  070204                  .WORD   .$.
12698  031656                          SBR     6$                      ;LOOP ON ERROR...
12699          000120                  MICPC=MICPC+1
12700  031656  100477                  .WORD   .$.
12701  031660                  9$:     CALL    SCP1            ;SCOPE THE ERROR...
12702  031660                          MOVE    # <MICPC+3>,BREG
12703          000121                  MICPC=MICPC+1
12704  031660  000523                  .WORD   .$.
12705  031662                          SBR     SCP1
12706          000122                  MICPC=MICPC+1
12707  031662  104427                  .WORD   .$.
12708  031664                          MOVE    SPAD <4>,MLR    ;RESTORE DATA POINTER...
12709          000123                  MICPC=MICPC+1
12710  031664  070204                  .WORD   .$.
12711  031666                          SBR     6$              ;SCOPE THE DATA...
12712          000124                  MICPC=MICPC+1
12713  031666  100477                  .WORD   .$.
12714  031670                          MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
12715          000125                  MICPC=MICPC+1
```

```
12716  031670  000404                .WORD   .$
12717  031672                        $ADD    BREG,SPAD (4),MARINC    ;ALSO DATA POINTER.
12718  031672  077004                .WORD   .$.!MARINC!.DO
12719  031674                        $DEC    SPAD (7)                ;IS IT DONE??
12720                  000126         MICPC=MICPC+1
12721  031674  063167          .WORD   .$.!.DSP
12722  031676                        BZ      4$                      ;YES, SCOPE THE TEST.
12723                  000127         MICPC=MICPC+1
12724  031676  101532                .WORD   .$.
12725  031700                        $BR     2$                      ;DO, THE NEXT.
12726                  000130         MICPC=MICPC+1
12727  031700  100453                .WORD   .$.
12728  031702                4$:     CALL    $CPE                    ;SCOPE THE TEST...
12729  031702                        MOVE    # (MICPC+3),BREG
12730                  000131         MICPC=MICPC+1
12731  031702  000533                .WORD   .$.
12732  031704                        $BR     $CPE
12733                  000132         MICPC=MICPC+1
12734  031704  104454                .WORD   .$.
12735  031706                        $BR     1$                      ;DO THE NEXT ITERATION...
12736                  000133         MICPC=MICPC+1
12737  031706  100400                .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                SXZ
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                SXZ
12750                                 ;:**********************************************************************
12751
12752  031710                STSTN
12753                                 ;  TEST 57
12754                                 ;--------------
12755  031710  012737  000057  001202  TST57:  MOV     #57,$TSTNM              ; LOAD THE N      THIS TEST
12756  031716  012737  032236  001442          MOV     #TST60,NEXT             ; POINT TO T      RT OF NEXT TEST.
12757                                                          ;R1 CONTAINS BASE KM...  ADDRESS
12758  031724  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
12759  031730  031744                          MCT57
12760  031732  104022                          ERROR   22                      ;TIME OUT ERROR...
12761  031734  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
12762  031740  000177  147476                  JMP     @NEXT                   ;GO TO NEXT TEST...
12763  031744                        MCT57:
12764  031744                        1$:     MOVE    # 0,MLR          ;SET MAR<-LO.
12765                  000000                MICPC=MICPC+1
12766  031744  010000                .WORD   .$.
12767  031746                        MOVE    # 0,MPR          ;SET MAR<-HI.
12768                  000001                MICPC=MICPC+1
12769  031746  004000                .WORD   .$.
12770  031750                        MOVE    # 0,BREG                 ;
12771                  000002                MICPC=MICPC+1
```

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

```
12828                   000025
12829 032016 016777
12830 032020
1283:                   000026
12832 032020 016777
12833 032022
12834                   000027
12835 032022 016777
12836 032024
12837                   000030
12838 032024 016400
12839 032026
12840                   000031
12841 032026 016525
12842 032030
12843                   000032
12844 032030 016525
12845 032032
12846                   000033
12847 032032 016525
12848 032034
12849                   000034
12850 032034 016526
12851 032036
12852                   000035
12853 032036 016652
12854 032040
12855                   000036
12856 032040 016525
12857 032042
12858                   000037
12859 032042 016652
12860 032044
12861                   000040
12862 032044 016653
12863 032046
12864                   000041
12865 032046 016525
12866 032050
12867                   000042
12868 032050 016652
12869 032052
12870                   000043
12871 032052 016525
12872 032054
12873                   000044
12874 032054 016526
12875 032056
12876                   000045
12877 032056 016652
12878 032060
12879                   000046
12880 032060 016652
12881 032062
12882                   000047
12883 032062 016652
```

```
MICPC=MICPC+1
.WORD   .S.
MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .S.
MOVE    # -1,MEM MARINC
MICPC=MICPC+1
.WORD   .S.
MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 125,MEM MARINC
MICPC=MICPC+1
.WORD   .S.
MOVE    # 126,MEM MARINC         ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 253,MEM MARINC         ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY
MICPC=MICPC+1
.WORD   .S.
MOVE    # 125,MEM MARINC
MICPC=MICPC+1
.WORD   .S.
MOVE    # 126,MEM MARINC         ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .S.
```

```
12884  032064                          MOVE    # 253,MEM MARINC         ;RESULT WITH C BIT SET.
12885          000050                  MICPC=MICPC+1
12886  032064  016653                  .WORD   .$.
12887  032066                          MOVE    # 7,SPAD <7>     ;SET THE COUNT.
12888          000051                  MICPC=MICPC+1
12889  032066  003007                  .WORD   .$.
12890
12891  032070                          MOVE    # 0,MLR          ;MAR←0.
12892          000052                  MICPC=MICPC+1
12893  032070  010000                  .WORD   .$.
12894
12895  032072                  2$:     MOVE    # 0,BREG                  ;
12896          000053                  MICPC=MICPC+1
12897  032072  000400                  .WORD   .$.
12898  032074                          SADD    SPAD <1>,BREG    ;CLEAR C BIT.
12899          000054                  MICPC=MICPC+1
12900  032074  060401                  .WORD   .$.!.DD
12901  032076                          MOVE    MEM,SPAD <0> MARINC      ;GET THE FIRST OPERAND.
12902          000055                  MICPC=MICPC+1
12903  032076  057220                  .WORD   .$.
12904  032100                          SADC    SPAD <0>,BR.SP,MARINC    ;
12905          000056                  MICPC=MICPC+1
12906  032100  077500                  .WORD   .$.
12907  032102                          SIFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
12908
12909
12910  032102                          SU32C   SPAD <0>,MEM,NOP
12911          000057                  MICPC=MICPC+1
12912  032102  040360                  .WORD   .$.
12913  032104                          BZ      3$
12914          000060                  MICPC=MICPC+1
12915  032104  101473                  .WORD   .$.
12916  032106                          MOVE    MEM,OUT1 <4>     ;GOOD DATA
12917          000061                  MICPC=MICPC+1
12918  032106  041224                  .WORD   .$.
12919  032110                          MOVE    BREG,OUT1 <5>    ;BAD DATA.
12920          000062                  MICPC=MICPC+1
12921  032110  061225                  .WORD   .$.
12922  032112                          MOVE    # 15,BREG        ;SET TYPE OF ERROR.
12923          000063                  MICPC=MICPC+1
12924  032112  000415                  .WORD   .$.
12925  032114                          MOVE    BREG,OUT1 <3>    ;SET TYPE OF ERROR.
12926          000064                  MICPC=MICPC+1
12927  032114  061223                  .WORD   .$.
12928  032116                          MOVE    # 04,BREG        ;LOAD FUNCTION CODE...
12929          000065                  MICPC=MICPC+1
12930  032116  000404                  .WORD   .$.
12931  032120                          MOVE    BREG,OUT1 <CSR7>         ;LOAD IT...
12932          000066                  MICPC=MICPC+1
12933  032120  061227                  .WORD   .$.
12934  032122                          CALL    EROR             ;ALU A PLUS C ERROR...
12935  032122                          MOVE    # <MICPC+3>,BREG
12936          000067                  MICPC=MICPC+1
12937  032122  000471                  .WORD   .$.
12938  032124                          SBR     EROR
12939          000070                  MICPC=MICPC+1
```

```
032124  104400          .WORD   .$.
032126                  MOVE    SPAD (4),MLR    ;RESET DATA POINTER...
        000071          MICPC=MICPC+1
032126  070204          .WORD   .$.
032130                  SBR     2$              ;LOOP ON ERROR...
        000072          MICPC=MICPC+1
032130  100453          .WORD   .$.
032132          3$:     CALL    SCP1
032132                  MOVE    # <MICPC+3>,BREG
        000073          MICPC=MICPC+1
032132  000475          .WORD   .$.
032134                  SBR     SCP1
        000074          MICPC=MICPC+1
032134  104427          .WORD   .$.
032136                  MOVE    SPAD (4),MLR    ;
        000075          MICPC=MICPC+1
032136  070204          .WORD   .$.
032140                  SBR     2$              ; SCOPE THE DATA....
        000076          MICPC=MICPC+1
032140  100453          .WORD   .$.
032142          6$:     MOVE    SPAD (4),MLR    ;RESET DATA POINTER...
        000077          MICPC=MICPC+1
032142  070204          .WORD   .$.
032144                  MOVE    MEM,SPAD (0),MARINC     ;GET FIRST OPRAND...
        000100          MICPC=MICPC+1
032144  057220          .WORD   .$.
032146                  MOVE    # 377,BREG      ;
        000101          MICPC=MICPC+1
032146  000777          .WORD   .$.
032150                  SADD    SPAD (2),BREG   ;SET C BIT...
        000102          MICPC=MICPC+1
032150  060402          .WORD   .$.!.DD
032152                  SADC    SPAD (0),BR.SP,MARINC           ;SP 0 & BR = A PLUS C
        000103          MICPC=MICPC+1
032152  077500          .WORD   .$.
032154                  MOVE    MEM,SPAD (3),MARINC     ;DUMMY INSTR, TO MARINC.
        000104          MICPC=MICPC+1
032154  057223          .WORD   .$.
032156                  $IFEQ   [illegible]     9$      ;BR IF GOOD...


032156                  SUB2C   SPAD (3),MEM,NOP
        000105          MICPC=MICPC+1
032156  040360          .WORD   .$.
032160                  BZ      9$
        000106          MICPC=MICPC+1
032160  101521          .WORD   .$.
032162                  MOVE    MEM,OUT1 (CSR4) ;GOOD DATA.
        000107          MICPC=MICPC+1
032162  041224          .WORD   .$.
032164                  MOVE    BREG,OUT1 (CSR5)        ;BAD DATA.
        000110          MICPC=MICPC+1
032164  061225          .WORD   .$.
032166                  MOVE    # 23,BREG               ;ERROR TYPE...
        000111          MICPC=MICPC+1
032166  000423          .WORD   .$.
```

```
12996 032170
12997         000112
12998 032170  061223
12999 032172
13000         000113
13001 032172  000404
13002 032174
13003         000114
13004 032174  061227
13005 032176
13006 032176
13007         000115
13008 032176  000517
13009 032200
13010         000116
13011 032200  104400
13012 032202
13013         000117
13014 032202  070204
13015 032204
13016         000120
13017 032204  100477
13018 032206                    95:     CALL    SCP1            ;SCOPE THE ERROR...
13019 032206                            MOVE    # <MICPC+3>,BREG
13020         000121                    MICPC=MICPC+1
13021 032206  000523                    .WORD   .$.
13022 032210                            SBR     SCP1
13023         000122                    MICPC=MICPC+1
13024 032210  104427                    .WORD   .$.
13025 032212                            MOVE    SPAD <4>,MLR    ;RESTORE DATA POINTER...
13026         000123                    MICPC=MICPC+1
13027 032212  070204                    .WORD   .$.
13028 032214                            SBR     6$              ;SCOPE THE DATA...
13029         000124                    MICPC=MICPC+1
13030 032214  100477                    .WORD   .$.
13031 032216                            MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
13032         000125                    MICPC=MICPC+1
13033 032216  000404                    .WORD   .$.
13034 032220                            SADD    BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
13035 032220  077004                    .WORD   .$.!MARINC!.DD
13036 032222                            SDEC    SPAD <7>                ;IS IT DONE??
13037         000126                    MICPC=MICPC+1
13038 032222  063167            .WORD   .$.!.DSP
13039 032224                            BZ      4$                      ;YES, SCOPE THE TEST.
13040         000127                    MICPC=MICPC+1
13041 032224  101532                    .WORD   .$.
13042 032226                            SBR     2$                      ;DO, THE NEXT.
13043         000130                    MICPC=MICPC+1
13044 032226  100453                    .WORD   .$.
13045 032230                    4$:     CALL    SCPE            ;SCOPE THE TEST...
13046 032230                            MOVE    # <MICPC+3>,BREG
13047         000131                    MICPC=MICPC+1
13048 032230  000533                    .WORD   .$.
13049 032232                            SBR     SCPE
13050         000132                    MICPC=MICPC+1
13051 032232  104454                    .WORD   .$.
                                        MOVE    BREG,OUT1 <CSR3>        ;
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    # 0A,BREG               ;ALU FUNCTION CODE.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        CALL    ERROR   ;REPORT ERROR...
                                        MOVE    # <MICPC+3>,BREG
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        SBR     ERROR
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        SBR     6$                      ;LOOP ON ERROR...
                                        MICPC=MICPC+1
                                        .WORD   .$.
```

```
13052 032234                            SBR     15              ;DO THE NEXT ITERATION...
13053         000133                    MICPC=MICPC+1
13054 032234  100400                    .WORD   .$.
13055 032236                    $ALUT1  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                    $XZ
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                    $XZ
13067                                   ;:*************************************************************
13068
13069 032236                    $TSTN
13070                                   ;  TEST 60
13071                                   ;-------------
13072 032236 012737 000060 001202 TST60: MOV     #60,$TSTNM              ; 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,LDVRWT               ;LOAD-VERIFY-WAIT.
13076 032256 032272                     MCT60
13077 032260 104022                     ERROR   22                      ;TIME OUT ERROR...
13078 032262 012706 001200              MOV     #STACK,SP               ;RESET STACK...
13079 032266 000177 147150              JMP     @NEXT                   ;GO TO NEXT TEST...
13080 032272                    MCT60:
13081 032272                    1$:     MOVE    # 0,MLR          ;SET MAR+LO.
13082         000000                    MICPC=MICPC+1
13083 032272  010000                    .WORD   .$.
13084 032274                            MOVE    # 0,MPR          ;SET MAR+HI.
13085         000001                    MICPC=MICPC+1
13086 032274  004000                    .WORD   .$.
13087 032276                            MOVE    # 0,BREG                 ;
13088         000002                    MICPC=MICPC+1
13089 032276  000400                    .WORD   .$.
13090 032300                            MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS...
13091         000003                    MICPC=MICPC+1
13092 032300  063236                    .WORD   .$.
13093 032302                            MOVE    BREG,SPAD <0>   ;
13094         000004                    MICPC=MICPC+1
13095 032302  063220                    .WORD   .$.
13096 032304                            MOVE    BREG,SPAD <1>   ;
13097         000005                    MICPC=MICPC+1
13098 032304  063221                    .WORD   .$.
13099 032306                            MOVE    BREG,SPAD <2>   ;
13100         000006                    MICPC=MICPC+1
13101 032306  063222                    .WORD   .$.
13102 032310                            SDEC    SPAD <2>                ;
13103         000007                    MICPC=MICPC+1
13104 032310  063162            .WORD   .$.!.DSP
13105 032312                            MOVE    BREG,SPAD <4>           ;
13106         000010                    MICPC=MICPC+1
13107 032312  063224                    .WORD   .$.
```

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

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

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

```
13276  032466  100453          .WORD   .$
13277  032470                  6$:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
13278          000077                  MICPC=MICPC+1
13279  032470  070204                  .WORD   .$.
13280  032472                          MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
13281          000100                  MICPC=MICPC+1
13282  032472  057220                  .WORD   .$.
13283  032474                          MOVE    # 377,BREG      ;
13284          000101                  MICPC=MICPC+1
13285  032474  000777                  .WORD   .$.
13286  032476                          SADD    SPAD <2>,BREG   ;SET C BIT...
13287          000102                  MICPC=MICPC+1
13288  032476  060402                  .WORD   .$.!.DO
13289  032500                          SUB2C   MEM,SPAD <0>,BR.SP,MARINC       ;SP 0 & BR = 2'S COMP SUB
13290          000103                  MICPC=MICPC+1
13291  032500  057760                  .WORD   .$.
13292  032502                          MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
13293          000104                  MICPC=MICPC+1
13294  032502  057223                  .WORD   .$.
13295  032504                          SIFEQ   MEM,SPAD <C>    9$      ;BR IF GOOD...
13296
13297
13298  032504                          SUB2C   SPAD <0>,MEM,NOP
13299          000105                  MICPC=MICPC+1
13300  032504  040360                  .WORD   .$.
13301  032506                          BZ      6$
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   .$.
13322  032524                          CALL    ERROR   ;REPORT ERROR...
13323  032524                          MOVE    # <MICPC+3>,BREG
13324          000115                  MICPC=MICPC+1
13325  032524  000517                  .WORD   .$.
13326  032526                          SBR     ERROR
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
13333                 000120
13334  032532         100477
13335  032534
13336  032534
13337                 000121
13338  032534         000523
13339  032536
13340                 000122
13341  032536         104427
13342  032540
13343                 000123
13344  032540         070204
13345  032542
13346                 000124
13347  032542         100477
13348  032544
13349                 000125
13350  032544         000404
13351  032546
13352  032546         077004
13353  032550
13354                 000126
13355  032550         063167
13356  032552
13357                 000127
13358  032552         101532
13359  032554
13360                 000130
13361  032554         100453
13362  032556
13363  032556
13364                 000131
13365  032556         000533
13366  032560
13367                 000132
13368  032560         104454
13369  032562
13370                 000133
13371  032562         100400
13372  032564
13373  032564
13374
13375
13376
13377
13378
13379
13380
13381
13382
13383  032564
13384
13385
13386  032564
13387
```

```
                SBR     6$                      ;LOOP ON ERROR...
                MICPC=MICPC+1
                .WORD   .$.
95:             CALL    SCP1                    ;SCOPE THE ERROR...
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     SCP1
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     6$                      ;SCOPE THE DATA...
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 4,BREG                 ;UPDATE BACKGROUND POINTER.
                MICPC=MICPC+1
                .WORD   .$.
                SADD    BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
                .WORD   .$.!MARINC!.DO
                $DEC    SPAD <7>                ;IS IT DONE??
                MICPC=MICPC+1
.WORD           .$.!.DSP
                BZ      4$                      ;YES, SCOPE THE TEST.
                MICPC=MICPC+1
                .WORD   .$.
                SBR     2$                      ;DO, THE NEXT.
                MICPC=MICPC+1
                .WORD   .$.
4$:             CALL    SCPE                    ;SCOPE THE TEST...
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     SCPE
                MICPC=MICPC+1
                .WORD   .$.
                SBR     1$                      ;DO THE NEXT ITERATION...
                MICPC=MICPC+1
                .WORD   .$.
$ALUT1  0,<DEC A>,$DEC,-1,-1,376,376,-1,-1,376,376,124,124,251,251,124,124,251,251,<A-1>
$XZ


                ;************************** 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..
$XZ
                ;;***************************************************************

$TSTN
                ;  TEST 61
```

```
                                .--------------
032564 012737 000061 001202 TST61:  MOV     #61,$TSTNM              ; LOAD THE NO. OF THIS TEST
032572 012737 033112 001442         MOV     #TST62,NEXT             ; POINT TO THE START OF NEXT TEST.
                            ;R1 CONTAINS BASE KMC11 ADDRESS
032600 004737 035536                JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
032604 032620                       MCT61
032606 104022                       ERROR   22                      ;TIME OUT ERROR...
032610 012706 001200                MOV     #STACK,SP               ;RESET STACK...
032614 000177 146622                JMP     @NEXT                   ;GO TO NEXT TEST...
032620                      MCT61:
032620                      1$:     MOVE    # 0,MLR          ;SET MAR←LO.
000000                              MICPC=MICPC+1
032620 010000                       .WORD   .$.
032622                              MOVE    # 0,MPR          ;SET MAR←HI.
000001                              MICPC=MICPC+1
032622 004000                       .WORD   .$.
032624                              MOVE    # 0,BREG                 ;
000002                              MICPC=MICPC+1
032624 000400                       .WORD   .$.
032626                              MOVE    BREG,SPAD (16)          ;FOR RETURN ADDRESS...
000003                              MICPC=MICPC+1
032626 063236                       .WORD   .$.
032630                              MOVE    BREG,SPAD (0)   ;
000004                              MICPC=MICPC+1
032630 063220                       .WORD   .$.
032632                              MOVE    BREG,SPAD (1)   ;
000005                              MICPC=MICPC+1
032632 063221                       .WORD   .$.
032634                              MOVE    BREG,SPAD (2)   ;
000006                              MICPC=MICPC+1
032634 063222                       .WORD   .$.
032636                              SPEC    SPAD (2)                ;
000007                              MICPC=MICPC+1
032636 063162               .WORD   .$.!.DSP
032640                              MOVE    BREG,SPAD (4)           ;
000010                              MICPC=MICPC+1
032640 063224                       .WORD   .$.
032642                              MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
000011                              MICPC=MICPC+1
032642 016400                       .WORD   .$.
032644                              MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEM
000012                              MICPC=MICPC+1
032644 016400                       .WORD   .$.
032646                              MOVE    # -1,MEM MARINC
000013                              MICPC=MICPC+1
032646 016777                       .WORD   .$.
032650                              MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
000014                              MICPC=MICPC+1
032650 016777                       .WORD   .$.
032652                              MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
000015                              MICPC=MICPC+1
032652 016777                       .WORD   .$.
032654                              MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
000016                              MICPC=MICPC+1
032654 016400                       .WORD   .$.
032656                              MOVE    # 376,MEM MARINC
```

```
13444                   000017
13445 032654 016776             MICPC=MICPC+1
13446 032660                    .WORD   .$.
13447                   000020  MOVE    # 376,MEM MARINC        ;RESULT WITH C BIT SET.
13448 032660 016776             MICPC=MICPC+1
13449 032662                    .WORD   .$.
13450                   000021  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
13451 032662 016400             MICPC=MICPC+1
13452 032664                    .WORD   .$.
13453                   000022  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
13454 032664 016777             MICPC=MICPC+1
13455 032666                    .WORD   .$.
13456                   000023  MOVE    # -1,MEM MARINC
13457 032666 016777             MICPC=MICPC+1
13458 032670                    .WORD   .$.
13459                   000024  MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
13460 032670 016777             MICPC=MICPC+1
13461 032672                    .WORD   .$.
13462                   000025  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
13463 032672 016777             MICPC=MICPC+1
13464 032674                    .WORD   .$.
13465                   000026  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
13466 032674 016777             MICPC=MICPC+1
13467 032676                    .WORD   .$.
13468                   000027  MOVE    # 376,MEM MARINC
13469 032676 016776             MICPC=MICPC+1
13470 032700                    .WORD   .$.
13471                   000030  MOVE    # 376,MEM MARINC         ;RESULT WITH C BIT SET.
13472 032700 016776             MICPC=MICPC+1
13473 032702                    .WORD   .$.
13474                   000031  MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
13475 032702 016525             MICPC=MICPC+1
13476 032704                    .WORD   .$.
13477                   000032  MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
13478 032704 016525             MICPC=MICPC+1
13479 032706                    .WORD   .$.
13480                   000033  MOVE    # 124,MEM MARINC
13481 032706 016524             MICPC=MICPC+1
13482 032710                    .WORD   .$.
13483                   000034  MOVE    # 124,MEM MARINC         ;RESULT WITH C BIT SET.
13484 032710 016524             MICPC=MICPC+1
13485 032712                    .WORD   .$.
13486                   000035  MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
13487 032712 016652             MICPC=MICPC+1
13488 032714                    .WORD   .$.
13489                   000036  MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
13490 032714 016525             MICPC=MICPC+1
13491 032716                    .WORD   .$.
13492                   000037  MOVE    # 251,MEM MARINC
13493 032716 016651             MICPC=MICPC+1
13494 032720                    .WORD   .$.
13495                   000040  MOVE    # 251,MEM MARINC         ;RESULT WITH C BIT SET.
13496 032720 016651             MICPC=MICPC+1
13497 032722                    .WORD   .$.
13498                   000041  MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
13499 032722 016525             MICPC=MICPC+1
                                .WORD   .$.
```

```
13500 032724
13501         000042
13502 032724  016652
13503 032726
13504         000043
13505 032726  016524
13506 032730
13507         000044
13508 032730  016524
13509 032732
13510         000045
13511 032732  016652
13512 032734
13513         000046
13514 032734  016652
13515 032736
13516         000047
13517 032736  016651
13518 032740
13519         000050
13520 032740  016651
13521 032742
13522         000051
13523 032742  003007
13524
13525 032744
13526         000052
13527 032744  010000
13528
13529 032746
13530         000053
13531 032746  000400
13532 032750
13533         000054
13534 032750  060401
13535 032752
13536         000055
13537 032752  057220
13538 032754
13539         000056
13540 032754  077560
13541 032756
13542
13543
13544 032756
13545         000057
13546 032756  0403E0
13547 032760
13548         000060
13549 032760  101473
13550 032762
13551         000061
13552 032762  041224
13553 032764
13554         000062
13555 032764  061225
```

```
        MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 124,MEM MARINC
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 124,MEM MARINC         ;RESULT WITH C BIT SET.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 251,MEM MARINC         ;LOAD THE DATA IN MEMORY.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 251,MEM MARINC         ;RESULT WITH C BIT SET.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 7,SPAD <7>     ;SET THE COUNT.
        MICPC=MICPC+1
        .WORD   .$.

        MOVE    # 0,MLR          ;MAR+0.
        MICPC=MICPC+1
        .WORD   .$.

2$:     MOVE    # 0,BREG                  ;
        MICPC=MICPC+1
        .WORD   .$.
        $ADD    SPAD <1>,BREG   ;CLEAR C BIT.
        MICPC=MICPC+1
        .WORD   .$.!.DD
        MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
        MICPC=MICPC+1
        .WORD   .$.
        $DEC    SPAD <0>,BR.SP,MARINC   ;
        MICPC=MICPC+1
        .WORD   .$.
        $IFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.


        SUB2C   SPAD <0>,MEM,NOP
        MICPC=MICPC+1
        .WORD   .$.
        BZ      3$
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,OUT1 <4>    ;GOOD DATA
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 <5>   ;BAD DATA.
        MICPC=MICPC+1
        .WORD   .$.
```

```
13556 032766
13557         000063
13560 032766  000415
13561 032770
13562         000064
13563 032770  061223
13564 032772
13565         000065
13566 032772  000407
13567 032774
13570         000066
13571 032774  061227
13572 032776
13573         000067
13574 032776  000471
13575 033000
13576         000070
13577 033000  104400
13600 033002
13601         000071
13602 033002  070204
13603 033004
13604         000072
13605 033004  100453
13606 033006
13607         000073
13610 033006  000475
13611 033010
13612         000074
13613 033010  104427
13614 033012
13615         000075
13616 033012  070204
13617 033014
13620         000076
13621 033014  100453
13622 033016
13623         000077
13624 033016  070204
13625 033020
13626         000100
13627 033020  057220
13630 033022
13631         000101
13632 033022  000777
13633 033024
13634         000102
13635 033024  060402
13636 033026
13637         000103
13640 033026  077560
13641 033030
13642         000104
13643 033030  057223
```

```
        MOVE    # 15,BREG        ;SET TYPE OF ERROR.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 07,BREG       ;LOAD FUNCTION CODE...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
        MICPC=MICPC+1
        .WORD   .$.
        CALL    ERROR           ;ALU DEC A ERROR...
        MOVE    # <MICPC+3>,BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     ERROR
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
        MICPC=MICPC+1
        .WORD   .$.
        SBR     2$              ;LOOP ON ERROR...
        MICPC=MICPC+1
        .WORD   .$.
3$:     CALL    SCP1
        MOVE    # <MICPC+3>,BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     SCP1
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD <4>,MLR    ;
        MICPC=MICPC+1
        .WORD   .$.
        SBR     2$              ; SCOPE THE DATA....
        MICPC=MICPC+1
        .WORD   .$.
6$:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 377,BREG      ;
        MICPC=MICPC+1
        .WORD   .$.
        SADD    SPAD <2>,BREG   ;SET C BIT...
        MICPC=MICPC+1
        .WORD   .$.!.DD
        SDEC    SPAD <0>,BR.SP,MARINC           ;SP 0 & BR = DEC A
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
        MICPC=MICPC+1
        .WORD   .$.
```

```
13612  033032                          SIFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...
13613
13614
13615  033032                          SUB2C   SPAD <0>,MEM,NOP
13616         000105                   MICPC=MICPC+1
13617  033032 040360                   .WORD   .$.
13618  033034                          BZ      9$
13619         000106                   MICPC=MICPC+1
13620  033034 101521                   .WORD   .$.
13621  033036                          MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
13622         000107                   MICPC=MICPC+1
13623  033036 041224                   .WORD   .$.
13624  033040                          MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
13625         000110                   MICPC=MICPC+1
13626  033040 061225                   .WORD   .$.
13627  033042                          MOVE    # 23,BREG               ;ERROR TYPE...
13628         000111                   MICPC=MICPC+1
13629  033042 000423                   .WORD   .$.
13630  033044                          MOVE    BREG,OUT1 <CSR3>        ;
13631         000112                   MICPC=MICPC+1
13632  033044 061223                   .WORD   .$.
13633  033046                          MOVE    # 07,BREG               ;ALU FUNCTION CODE.
13634         000113                   MICPC=MICPC+1
13635  033046 000407                   .WORD   .$.
13636  033050                          MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
13637         000114                   MICPC=MICPC+1
13638  033050 061227                   .WORD   .$.
13639  033052                          CALL    ERROR   ;REPORT ERROR...
13640  033052                          MOVE    # <MICPC+3>,BREG
13641         000115                   MICPC=MICPC+1
13642  033052 000517                   .WORD   .$.
13643  033054                          SBR     ERROR
13644         000116                   MICPC=MICPC+1
13645  033054 104400                   .WORD   .$.
13646  033056                          MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER.
13647         000117                   MICPC=MICPC+1
13648  033056 070204                   .WORD   .$.
13649  033060                          SBR     6$                      ;LOOP ON ERROR...
13650         000120                   MICPC=MICPC+1
13651  033060 100477                   .WORD   .$.
13652  033062                  9$:     CALL    SCP1            ;SCOPE THE ERROR...
13653  033062                          MOVE    # <MICPC+3>,BREG
13654         000121                   MICPC=MICPC+1
13655  033062 000523                   .WORD   .$.
13656  033064                          SBR     SCP1
13657         000122                   MICPC=MICPC+1
13658  033064 104427                   .WORD   .$.
13659  033066                          MOVE    SPAD <4>,MLR    ;RESTORE DATA POINTER...
13660         000123                   MICPC=MICPC+1
13661  033066 070204                   .WORD   .$.
13662  033070                          SBR     6$              ;SCOPE THE DATA...
13663         000124                   MICPC=MICPC+1
13664  033070 100477                   .WORD   .$.
13665  033072                          MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
13666         000125                   MICPC=MICPC+1
13667  033072 000404                   .WORD   .$.
```

```
13668  033074                                  SADD    BREG,SPAD (4),MARINC    ;ALSO DATA POINTER.
13669  033074  077004                          .WORD   .$.!MARINC!.DD
13670  033076                                  SDEC    SPAD (7)                ;IS IT DONE??
13671          000126                          MICPC=MICPC+1
13672  033076  063167                  .WORD   .$.!.DSP
13673  033100                                  BZ      4$                      ;YES, SCOPE THE TEST.
13674          000127                          MICPC=MICPC+1
13675  033100  101532                          .WORD   .$.
13676  033102                                  SBR     2$                      ;DO, THE NEXT.
13677          000130                          MICPC=MICPC+1
13678  033102  100453                          .WORD   .$.
13679  033104                          4$:     CALL    SCPE                    ;SCOPE THE TEST...
13680  033104                                  MOVE    @ (MICPC+3),BREG
13681          000131                          MICPC=MICPC+1
13682  033104  000533                          .WORD   .$.
13683  033106                                  SBR     SCPE
13684          000132                          MICPC=MICPC+1
13685  033106  104454                          .WORD   .$.
13686  033110                                  SBR     1$                      ;DO THE NEXT ITERATION...
13687          000133                          MICPC=MICPC+1
13688  033110  100400                          .WORD   .$.
13689  033112                          SPGMCK
13690  033112                          SXZ
13691
13692
13693                                          ;************************** TEST 62 ****************************
13694                                          ;*TEST OF PROGRAM CLOCK BIT
13695                                          ;*DO A MASTER CLEAR, VERIFY THAT PROGRAM CLOCK IS SET
13696                                          ;*WRITE PROGRAM CLOCK BIT TO A ONE, VERIFY THAT IT CLEARS.
13697                                          ;*AND THEN SETS SOME TIME LATER.
13698  033112                          SXZ
13699                                          ;;***********************************************************************
13700
13701  033112                          STSTN
13702                                  ;  TEST 62
13703                                  ;-------------
13704  033112  012737  000062  001202  TST62:  MOV     #62,STSTNM              ; LOAD THE NO  OF THIS TEST
13705  033120  012737  033260  001442          MOV     #TST63,NEXT             ; POINT TO THE START OF NEXT TEST.
13706                                                                  ;R1 CONTAINS BASE KMC11 ADDRESS
13707  033126  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
13708  033132  033146                          MCT62
13709  033134  104022                          ERROR   22                      ;TIME OUT ERROR...
13710  033136  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
13711  033142  000177  146274                  JMP     @NEXT                   ;GO TO NEXT TEST...
13712  033146                          MCT62:
13713  033146                          1$:     MOVE    # 0,BREG         ;PREPARE
13714          000000                          MICPC=MICPC+1
13715  033146  000400                  .WORD   .$.
13716  033150                                  MOVE    BREG,SPAD (2)   ;       FOR
13717          000001                          MICPC=MICPC+1
13718  033150  063222                  .WORD   .$.
13719  033152                                  MOVE    BREG,SPAD (3)   ;               DELAY.
13720          000002                          MICPC=MICPC+1
13721  033152  063223                  .WORD   .$.
13722  033154                                  MOVE    BREG,SPAD (4)   ;
13723          000003                          MICPC=MICPC+1
```

```
13724  033154  063224          .WORD   .$.
13725  033156                  MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSE...
13726          000004          MICPC=MICPC+1
13727  033156  063236          .WORD   .$.
13730  033160                  MOVE    INP1 <CSR11>,BREG       ;CHECK IF PGM. CLK BIT SET.
13731          000005          MICPC=MICPC+1
13732  033160  120620          .WORD   .$.
13733  033162                  BPH     2$                      ;YES, GO RESET IT.
13734          000006          MICPC=MICPC+1
13735  033162  103014          .WORD   .$.
13736  033164                  MOVE    # 14,MEM                ;ERROR PROGRAM CLOCK BIT IS NOT SET.
13737          000007          MICPC=MICPC+1
13740  033164  002414          .WORD   .$.
13741  033166                  MOVE    MEM,OUT1 <CSR3>         ;ERROR TYPE...
13742          000010          MICPC=MICPC+1
13743  033166  041223          .WORD   .$.
13744  033170                  CALL    EROR1                   ;SET TYPE OF ERROR
13745  033170                  MOVE    # <MICPC+3>,BREG
13746          000011          MICPC=MICPC+1
13747  033170  000413          .WORD   .$.
13750  033172                  $BR     ERROR1
13751          000012          MICPC=MICPC+1
13752  033172  104401          .WORD   .$.
13753  033174                  $BR     1$                      ;LOOP ON ERROR.
13754          000013          MICPC=MICPC+1
13755  033174  100400          .WORD   .$.
13756  033176          2$:     MOVE    # 20,BREG               ;GET SET TO CLEAR
13757          000014          MICPC=MICPC+1
13760  033176  000420          .WORD   .$.
13761  033200                  MOVE    BREG,OUT1 <CSR11>       ;PROGRAM CLOCK BIT.
13762          000015          MICPC=MICPC+1
13763  033200  061231          .WORD   .$.
13764  033202                  MOVE    INP1 <CSR11>,BREG       ;SEE IF IT IS CLEARED.
13765          000016          MICPC=MICPC+1
13766  033202  120620          .WORD   .$.
13767  033204                  BPH     3$                      ;NO, REPORT ERROR.
13770          000017          MICPC=MICPC+1
13771  033204  103021          .WORD   .$.
13772  033206                  $BR     4$                      ;GO SEE IF IT SETS AGAIN.
13773          000020          MICPC=MICPC+1
13774  033206  100426          .WORD   .$.
13775  033210          3$:
13776  033210                  MOVE    # 14,MEM                ;TYPE
13777          000021          MICPC=MICPC+1
14000  033210  002414          .WORD   .$.
14001  033212                  MOVE    MEM,OUT1 <CSR3>         ;
14002          000022          MICPC=MICPC+1
14003  033212  041223          .WORD   .$.
14004  033214                  CALL    EROR1                   ;PGM. CLK BIT DOES NOT CLEAR.
14005  033214                  MOVE    # <MICPC+3>,BREG
14006          000023          MICPC=MICPC+1
14007  033214  000425          .WORD   .$.
14010  033216                  $BR     ERROR1
14011          000024          MICPC=MICPC+1
14012  033216  104401          .WORD   .$.
14013  033220                  $BR     1$                      ;START AGAIN...
```

```
13780                   000025
13781   033220  100400          MICPC=MICPC+1
13782   033222                  .WORD   .$.
13783                   000026  4$:     MOVE    INP1 (CSR11),BREG       ;IS IT SET AGAIN??
13784   033222  120620          MICPC=MICPC+1
13785   033224                  .WORD   .$.
13786                   000027          BR4     5$                      ;YES, IT'S OK.
13787   033224  103042          MICPC=MICPC+1
13788   033226                  .WORD   .$.
13789                   000030          $INC    SPAD (2)                ;IN DELAY!
13790   033226  063062          MICPC=MICPC+1
13791   033230          .WORD   .$.!.DSP
13792                   000031          $ADC    SPAD (3)                ;DELAY!
13793   033230  063103          MICPC=MICPC+1
13794   033232          .WORD   .$.!.DSP
13795                   000032          MOVE    SPAD (3),BREG   ;IS DELAYED ENOUGH?
13796   033232  060603          MICPC=MICPC+1
13797   033234                  .WORD   .$.
13798                   000033          BR7     6$                      ;YES, REPORT ERROR.
13799   033234  103435          MICPC=MICPC+1
13800   033236                  .WORD   .$.
13801                   000034          $BR     4$                      ;CONTINUE DELAY LOOP.
13802   033236  100426          MICPC=MICPC+1
13803   033240                  .WORD   .$.
13804   033240          6$:
13805                   000035          MOVE    # 14,MEM                ;SET TYPE OF ERROR
13806   033240  002414          MICPC=MICPC+1
13807   033242                  .WORD   .$.
13808                   000036          MOVE    MEM,OUT1 (CSR3)         ;
13809   033242  041223          MICPC=MICPC+1
13810   033244                  .WORD   .$.
13811   033244                  CALL    ERROR1                  ;ERROR PGM. CLK NOT SET.
13812                   000037          MOVE    # (MICPC+3),BREG
13813   033244  000441          MICPC=MICPC+1
13814   033246                  .WORD   .$.
13815                   000040          $BR     ERROR1
13816   033246  104401          MICPC=MICPC+1
13817   033250                  .WORD   .$.
13818                   000041          $BR     1$                      ;LOOP ON ERROR.
13819   033250  100400          MICPC=MICPC+1
13820   033252                  .WORD   .$.
13821   033252          5$:     CALL    SCPE                    ;SCOPE THE TEST.
13822                   000042          MOVE    # (MICPC+3),BREG
13823   033252  000444          MICPC=MICPC+1
13824   033254                  .WORD   .$.
13825                   000043          $BR     SCPE
13826   033254  104454          MICPC=MICPC+1
13827   033256                  .WORD   .$.
13828                   000044          $BR     1$                      ;DO NEXT PASS.
13829   033256  100400          MICPC=MICPC+1
13830   033260                  .WORD   .$.
13831   033260          $NOISE
13832                   $XZ
13833
13834                   ;*************************** TEST 63 ****************************
13835                   ;*MICRO-PROCESSOR NOISE TEST.
```

```
                                        ;#WRITE ALL ZERO'S THEN ALL ONE'S THEN A DATA PATTERN TO
                                        ;#THE IBUS*,IBUS,SP,S MAIN MEMORY
                                        ;#THEN GO AND READ BACK THE DATA PATTERN
                                        ;#TO VERIFY THAT READING AND WRITING OF OTHER
                                        ;#LOCATIONS DID NOT CHANGE DATA.
033260                          SXZ
                                        ;;*****************************************************************
033260                          STSTN
                                ;  TEST 63
                                ;-------------
033260  012737  000063  001202  TST63:  MOV     #63,$TSTNM              ; LOAD THE NO. OF THIS TEST
033266  012737  034064  001442          MOV     #TST64,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
033274  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
033300  033314                          MCT63
033302  104022                          ERROR   22                      ;TIME OUT ERROR...
033304  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
033310  000177  146126                  JMP     @NEXT                   ;GO TO NEXT TEST...
033314                          MCT63:
033314                          18$:    MOVE    # 0,BREG                 ;START WITH ZERO.
        000000                          MICPC=MICPC+1
033314  000400                          .WORD   .$.
033316                                  MOVE    BREG,SPAD <0>   ;
        000001                          MICPC=MICPC+1
033316  063220                          .WORD   .$.
033320                                  MOVE    BREG,SPAD <1>   ;
        000002                          MICPC=MICPC+1
033320  063221                          .WORD   .$.
033322                                  MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
        000003                          MICPC=MICPC+1
033322  063236                          .WORD   .$.

033324                          1$:     MOVE    # 0,MLR ;MAR←0.
        000004                          MICPC=MICPC+1
033324  010000                          .WORD   .$.
033326                                  MOVE    # 0,MPR ;
        000005                          MICPC=MICPC+1
033326  004000                          .WORD   .$.
033330                                  MOVE    BREG,OUT1 <CSR0>        ;WRITE IBUS* REGISTERS.
        000006                          MICPC=MICPC+1
033330  061220                          .WORD   .$.
033332                                  MOVE    BREG,OUT1 <CSR2>        ;WRITE IBUS* REGISTERS.
        000007                          MICPC=MICPC+1
033332  061222                          .WORD   .$.
033334                                  MOVE    BREG,OUT1 <CSR3>        ;WRITE IBUS* REGISTERS.
        000010                          MICPC=MICPC+1
033334  061223                          .WORD   .$.
033336                                  MOVE    BREG,OUT1 <CSR4>        ;WRITE IBUS* REGISTERS.
        000011                          MICPC=MICPC+1
033336  061224                          .WORD   .$.
033340                                  MOVE    BREG,OUT1 <CSR5>        ;WRITE IBUS* REGISTERS.
        000012                          MICPC=MICPC+1
033340  061225                          .WORD   .$.

033342                                  MOVE    BREG,OUT1 <CSR6>        ;WRITE IBUS* REGISTERS.
```

```
13892                   000013
13893   033342  061226
13894   033344
13895                   000014
13896   033344  061227
13897
13898
13899   033346
13900                   000015
13901   033346  062220
13902   033350
13903                   000016
13904   033350  062221
13905   033352
13906                   000017
13907   033352  062222
13908   033354
13909                   000020
13910   033354  062223
13911   033356
13912                   000021
13913   033356  062224
13914   033360
13915                   000022
13916   033360  062225
13917   033362
13918                   000023
13919   033362  062226
13920   033364
13921                   000024
13922   033364  062227
13923
13924
13925
13926   033366
13927                   000025
13928   033366  063223
13929   033370
13930                   000026
13931   033370  063224
13932   033372
13933                   000027
13934   033372  063225
13935   033374
13936                   000030
13937   033374  063226
13938   033376
13939                   000031
13940   033376  063227
13941   033400
13942                   000032
13943   033400  063230
13944   033402
13945                   000033
13946   033402  063231
13947   033404
```

```
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT1 (CSR7)        ;WRITE IBUS* REGISTERS.
                MICPC=MICPC+1
                .WORD   .$.
        ;:*
        ;:*
                MOVE    BREG,OUT0 (0)   ;WRITE IBUS REGISTERS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT0 (1)   ;WRITE IBUS REGISTERS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT0 (2)   ;WRITE IBUS REGISTERS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT0 (3)   ;WRITE IBUS REGISTERS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT0 (4)   ;WRITE IBUS REGISTERS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT0 (5)   ;WRITE IBUS REGISTERS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT0 (6)   ;WRITE IBUS REGISTERS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT0 (7)   ;WRITE IBUS REGISTERS.
                MICPC=MICPC+1
                .WORD   .$.
        ;:*
        ;:*
        ;:*
                                        ;WRITE SPAD LOCATIONS.
                MOVE    BREG,SPAD (3)   ;WRITE SPAD LOCATIONS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,SPAD (4)   ;WRITE SPAD LOCATIONS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,SPAD (5)   ;WRITE SPAD LOCATIONS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,SPAD (6)   ;WRITE SPAD LOCATIONS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,SPAD (7)   ;WRITE SPAD LOCATIONS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,SPAD (10)  ;WRITE SPAD LOCATIONS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,SPAD (11)  ;WRITE SPAD LOCATIONS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,SPAD (12)  ;WRITE SPAD LOCATIONS.
```

```
13948                   000034
13949  033404  063232
13950  033406
13951                   000035
13952  033406  063233
13953  033410
13954                   000036
13955  033410  063234
13956  033412
13957                   000037
13958  033412  063235
13959
13960
13961  033414
13962                   000040
13963  033414  076620
13964  033416
13965                   000041
13966  033416  063060
13967  033420
13968                   000042
13969  033420  063101
13970  033422
13971                   000043
13972  033422  063222
13973  033424
13974                   000044
13975  033424  000404
13976  033426
13977
13978
13979  033426
13980                   000045
13981  033426  060361
13982  033430
13983                   000046
13984  033430  101050
13985  033432
13986                   000047
13987  033432  100452
13988  033434
13989  033434
13990                   000050
13991  033434  060602
13992  033436
13993                   000051
13994  033436  100454
13995  033440
13996                   000052
13997  033440  060602
13998  033442
13999                   000053
14000  033442  100440
14001  033444
14002  033444
14003                   000054
```

```
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,SPAD <13>  ;WRITE SPAD LOCATIONS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,SPAD <14>  ;WRITE SPAD LOCATIONS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,SPAD <15>  ;WRITE SPAD LOCATIONS.
                MICPC=MICPC+1
                .WORD   .$.
;:*
;:*
2$:             MOVE    BREG,MEM MARINC ;WRITE MEMORY LOCATION.
                MICPC=MICPC+1
                .WORD   .$.
                $INC    SPAD <0>                ;UPDATE COUNT.
                MICPC=MICPC+1
.WORD           .$.!.DSP
                $ADC    SPAD <1>                ;UPDATE COUNT.
                MICPC=MICPC+1
.WORD           .$.!.DSP
                MOVE    BREG,SPAD <2>   ;IS IT DONE?
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 4,BREG                ;IS IT DONE?
                MICPC=MICPC+1
                .WORD   .$.
                $IFHIS  BREG,SPAD <1> 3$        ;NO, CONTINUE.


                SUB2C   SPAD <1>,BREG,NOP
                MICPC=MICPC+1
                .WORD   .$.
                BC      64$
                MICPC=MICPC+1
                .WORD   .$.
                $BR     3$
                MICPC=MICPC+1
                .WORD   .$.
64$:
                MOVE    SPAD <2>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                $BR     4$              ;YES, WRITE THE NEXT.
                MICPC=MICPC+1
                .WORD   .$.
3$:             MOVE    SPAD <2>,BREG   ;RESTORE BREG.
                MICPC=MICPC+1
                .WORD   .$.
                $BR     2$              ;WRITE NEXT MEMORY LOCATION
                MICPC=MICPC+1
                .WORD   .$.
4$:
                $DEC    SPAD <2>                ;WAS IT ZERO?
                MICPC=MICPC+1
```

```
14004  033444  063162          .WORD   .$.!.DSP
14005  033446                  BZ      5$                      ;YES, NOW LOAD 377.
14006          000055          MICPC=MICPC+1
14007  033446  101462          .WORD   .$.
14008  033450                  MOVE    BREG,SPAD (2)           ;WAS IT 377?
14009          000056          MICPC=MICPC+1
14010  033450  063222          .WORD   .$.
14011  033452                  $INC    SPAD (2)                        ;WAS IT 377?
14012          000057          MICPC=MICPC+1
14013  033452  063062  .WORD   .$.!.DSP
14014  033454                  BC      6$                      ;YES, NOW LOAD 252.
14015          000060          MICPC=MICPC+1
14016  033454  101064          .WORD   .$.
14017  033456                  SBR     9$                      ;DONE START READING.
14018          000061          MICPC=MICPC+1
14019  033456  100471          .WORD   .$.
14020  033460          5$:     MOVE    # 377,BREG              ;LOAD 377.
14021          000062          MICPC=MICPC+1
14022  033460  000777          .WORD   .$.
14023  033462                  SBR     7$                      ;
14024          000063          MICPC=MICPC+1
14025  033462  100465          .WORD   .$.
14026  033464          6$:     MOVE    # 252,BREG              ;LOAD 252
14027          000064          MICPC=MICPC+1
14028  033464  000652          .WORD   .$.
14029  033466          7$:
14030  033466                  MOVE    # 0,SPAD (0)            ;RESET SPAD 0.
14031          000065          MICPC=MICPC+1
14032  033466  003000          .WORD   .$.
14033  033470                  MOVE    # 1,SPAD (1)            ;RESET SPAD 1.
14034          000066          MICPC=MICPC+1
14035  033470  003001          .WORD   .$.
14036  033472                  $DEC    SPAD (1)                        ;RESET SPAD 1.
14037          000067          MICPC=MICPC+1
14038  033472  063161  .WORD   .$.!.DSP
14039  033474                  SBR     1$                      ;CONTINUE
14040          000070          MICPC=MICPC+1
14041  033474  100404          .WORD   .$.
14042                  ;:*
14043                  ;:*     NOW GO BACK AND READ THEM.
14044                  ;:*
14045  033476          9$:     MOVE    BREG,SPAD (2)           ;SET UP FOR CHECKING.
14046          000071          MICPC=MICPC+1
14047  033476  063222          .WORD   .$.
14048  033500                  MOVE    INP1 (CSR0),SPAD (1)    ;GET IBUS* REGISTER
14049          000072          MICPC=MICPC+1
14050  033500  123001          .WORD   .$.
14051  033502                  CALL    CMPRE                   ;VERIFY IT.
14052  033502                  MOVE    # (MICPC+3),BREG
14053          000073          MICPC=MICPC+1
14054  033502  000475          .WORD   .$.
14055  033504                  SBR     CMPRE
14056          000074          MICPC=MICPC+1
14057  033504  100650          .WORD   .$.
14058  033506                  MOVE    INP1 (CSR2),SPAD (1)    ;READ NEXT IBUS* REG.
14059          000075          MICPC=MICPC+1
```

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

```
14116                   000116
14117  033550           100650          MICPC=MICPC+1
14118                                   .WORD   .$.
14119                                   ;:*
14120  033552                           ;:*
14121                   000117          MOVE    INPO (0),SPAD (1)       ;GET IBUS REG.
14122  033552           023001          MICPC=MICPC+1
14123  033554                           .WORD   .$.
14124  033554                           CALL    CMPRE           ;VERIFY IT.
14125  033554                           MOVE    @ (MICPC+3),BREG
14126                   000120          MICPC=MICPC+1
14127  033554           000522          .WORD   .$.
14128  033556                           SBR     CMPRE
14129                   000121          MICPC=MICPC+1
14130  033560           100650          .WORD   .$.
14131  033560                           MOVE    INPO (1),SPAD (1)       ;GET NEXT IBUS-REGISTER.
14132                   000122          MICPC=MICPC+1
14133  033560           023021          .WORD   .$.
14134  033562                           CALL    CMPRE           ;VERIFY IT.
14135  033562                           MOVE    @ (MICPC+3),BREG
14136                   000123          MICPC=MICPC+1
14137  033562           000525          .WORD   .$.
14138  033564                           SBR     CMPRE
14139                   000124          MICPC=MICPC+1
14140  033564           100650          .WORD   .$.
14141  033566                           MOVE    INPO (2),SPAD (1)       ;GET NEXT IBUS-REG.
14142                   000125          MICPC=MICPC+1
14143  033566           023041          .WORD   .$.
14144  033570                           CALL    CMPRE           ;VERIFY IT.
14145  033570                           MOVE    @ (MICPC+3),BREG
14146                   000126          MICPC=MICPC+1
14147  033570           000530          .WORD   .$.
14148  033572                           SBR     CMPRE
14149                   000127          MICPC=MICPC+1
14150  033572           100650          .WORD   .$.
14151  033574                           MOVE    INPO (3),SPAD (1)       ;GET NEXT IBUS-REG.
14152                   000130          MICPC=MICPC+1
14153  033574           023061          .WORD   .$.
14154  033576                           CALL    CMPRE           ;VERIFY IT.
14155  033576                           MOVE    @ (MICPC+3),BREG
14156                   000131          MICPC=MICPC+1
14157  033576           000533          .WORD   .$.
14158  033600                           SBR     CMPRE
14159                   000132          MICPC=MICPC+1
14160  033600           100650          .WORD   .$.
14161  033602                           MOVE    INPO (4),SPAD (1)       ;GET NEXT IBUS-REG.
14162                   000133          MICPC=MICPC+1
14163  033602           023101          .WORD   .$.
14164  033604                           CALL    CMPRE           ;VERIFY IT.
14165  033604                           MOVE    @ (MICPC+3),BREG
14166                   000134          MICPC=MICPC+1
14167  033604           000536          .WORD   .$.
14168  033606                           SBR     CMPRE
14169                   000135          MICPC=MICPC+1
14170  033606           100650          .WORD   .$.
14171  033610                           MOVE    INPO (5),SPAD (1)       ;GET NEXT IBU
14172                   000136          MICPC=MICPC+1
```

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

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

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

```
14340                                   ;:*
14341                                   ;:*
14342 033756                            MOVE    # 0,MLR         ;
14343        000221                     MICPC=MICPC+1
14344 033756 010000                     .WORD   .$.
14345 033760                            MOVE    # 0,MPR         :
14346        000222                     MICPC=MICPC+1
14347 033760 004000                     .WORD   .$.
14348 033762                            MOVE    BREG,SPAD <0>   ;SET THE COUNTER.
14349        000223                     MICPC=MICPC+1
14350 033762 063220                     .WORD   .$.
14351 033764                            MOVE    # 3,SPAD <3>    ;SET THE COUNTER.
14352        000224                     MICPC=MICPC+1
14353 033764 003003                     .WORD   .$.
14354 033766                    125:    MOVE    MEM,BREG MARINC ;GET THE CONTENTS OF MEMORY LOCATION.
14355        000225                     MICPC=MICPC+1
14356 033766 054620                     .WORD   .$.
14357 033770                            $IFEQ   BREG,SPAD <2> 15$       ;BRANCH IF GOOD.
14358
14359
14360 033770                            SUB2C   SPAD <2>,BREG,NOP
14361        000226                     MICPC=MICPC+1
14362 033770 060362                     .WORD   .$.
14363 033772                            BZ      15$
14364        000227                     MICPC=MICPC+1
14365 033772 101637                     .WORD   .$.
14366 033774                            MOVE    MEM,OUT1 <CSR4> ;GOOD DATA
14367        000230                     MICPC=MICPC+1
14368 033774 041224                     .WORD   .$.
14369 033776                            MOVE    BREG,OUT1 <CSR5>        ;BAD DATA
14370        000231                     MICPC=MICPC+1
14371 033776 061225                     .WORD   .$.
14372 034000                            MOVE    # 20,BREG       ;
14373        000232                     MICPC=MICPC+1
14374 034000 000420                     .WORD   .$.
14375 034002                            MOVE    BREG,OUT1 <CSR3>        ;TYPE OF ERROR.
14376        000233                     MICPC=MICPC+1
14377 034002 061223                     .WORD   .$.
14378 034004                            CALL    ERROR1          ;DATA ERROR.
14379 034004                            MOVE    # <MICPC+3>,BREG
14380        000234                     MICPC=MICPC+1
14381 034004 000636                     .WORD   .$.
14382 034006                            SBR     ERROR1
14383        000235                     MICPC=MICPC+1
14384 034006 104401                     .WORD   .$.
14385 034010                            SBR     10$             ;LOOP ON ERROR.
14386        000236                     MICPC=MICPC+1
14387 034010 100400                     .WORD   .$.
14388 034012                    15$:    $INC    SPAD <0>                ;COUNT BY ONE.
14389        000237                     MICPC=MICPC+1
14390 034012 063060             .WORD   .$.!.DSP
14391 034014                            BC      16$             ;COUNT BY ONE.
14392        000240                     MICPC=MICPC+1
14393 034014 101242                     .WORD   .$.
14394 034016                            SBR     12$             ;DO THE NEXT.
14395        000241                     MICPC=MICPC+1
```

```
14396  034016  100625          .WORD   .$.
14397  034020                  16$:    SDEC    SPAD <3>                ;IS IT DONE??
14398          000242                  MICPC=MICPC+1
14399  034020  063163          .WORD   .$.!.DSP
14400  034022                          BZ      17$                     ;YES.SCOPE IT.
14401          000243                  MICPC=MICPC+1
14402  034022  101645                  .WORD   .$.
14403  034024                          SBR     16$                     ;ELSE, DO NEXT.
14404          000244                  MICPC=MICPC+1
14405  034024  100625                  .WORD   .$.
14406  034026                  17$:    CALL    SCPE                    ;SCOPE THE TEST...
14407  034026                          MOVE    # <MICPC+3),BREG
14408          000245                  MICPC=MICPC+1
14409  034026  000647                  .WORD   .$.
14410  034030                          SBR     SCPE
14411          000246                  MICPC=MICPC+1
14412  034030  104454                  .WORD   .$.
14413  034032                          SBR     10$                     ;DO THE NEXT ITERATION.
14414          000247                  MICPC=MICPC+1
14415  034032  100400                  .WORD   .$.
14416  034034                  CMPRE:  MOVE    BREG,SPAD <17>          ;SAVE THE RETURN ADDRESS.
14417          000250                  MICPC=MICPC+1
14418  034034  063237                  .WORD   .$.
14419  034036                          MOVE    SPAD <1>,BREG           ;GET FOUND IN BREG.
14420          000251                  MICPC=MICPC+1
14421  034036  060601                  .WORD   .$.
14422  034040                          SIFEQ   BREG,SPAD <2> 33$       ;BRANCH IF GOOD.
14423
14424
14425  034040                          SUB2C   SPAD <2>,BREG,NOP
14426          000252                  MICPC=MICPC+1
14427  034040  060362                  .WORD   .$.
14428  034042                          BZ      33$
14429          000253                  MICPC=MICPC+1
14430  034042  101663                  .WORD   .$.
14431  034044                          MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
14432          000254                  MICPC=MICPC+1
14433  034044  061225                  .WORD   .$.
14434  034046                          MOVE    SPAD <2>,BREG   ;
14435          000255                  MICPC=MICPC+1
14436  034046  060602                  .WORD   .$.
14437  034050                          MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA.
14438          000256                  MICPC=MICPC+1
14439  034050  061224                  .WORD   .$.
14440  034052                          MOVE    # 20,BREG       ;
14441          000257                  MICPC=MICPC+1
14442  034052  000420                  .WORD   .$.
14443  034054                          MOVE    BREG,OUT1 <CSR3>        ;TYPE OF ERROR.
14444          000260                  MICPC=MICPC+1
14445  034054  061223                  .WORD   .$.
14446  034056                          CALL    ERROR1          ;REPORT ERROR
14447  034056                          MOVE    # <MICPC+3),BREG
14448          000261                  MICPC=MICPC+1
14449  034056  000663                  .WORD   .$.
14450  034060                          SBR     ERROR1
14451          000262                  MICPC=MICPC+1
```

```
14452  034060  104401                      .WORD   .$.
14453  034062                      33$:
14454  034062                              $BR     SPAD (17) PAGED
14455          000263                       MICPC=MICPC+1
14456  034062  160617                       .WORD   .$.
14457  034064                      $HLRSR
14458  034064                      $XZ
14459
14460
14461                                       ;*************************** TEST 64 ****************************
14462                                       ;* HELL RAISER TEST...
14463                                       ;* ONLY TO TEST MPRAM CONTROL LOGIC...
14464                                       ;* NOT FOR MAINTENENCE PURPOSE...
14465  034064                      $XZ
14466                                       ;:*************************************************************
14467
14468  034064                      $TSTN
14469                                       ;  TEST 64
14470                                       ;-------------
14471  034064  012737  000064  001202  TST64:  MOV     #64,$TSTNM              ; LOAD THE NO. OF THIS TEST
14472  034072  012737  034314  001442          MOV     #TST65,NEXT             ; POINT TO THE START OF NEXT TEST
14473                                                                  ;R1 CONTAINS BASE KMC11 ADDRESS
14474  034100  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
14475  034104  034120                          MCT64
14476  034106  104022                          ERROR   22                      ;TIME OUT ERROR...
14477  034110  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
14478  034114  000177  145322                  JMP     @NEXT                   ;GO TO NEXT TEST...
14479  034120                      MCT64:
14480  034120                      1$:     MOVE    # 0,BREG         ;
14481          000000                      MICPC=MICPC+1
14482  034120  000400                      .WORD   .$.
14483  034122                              MOVE    # 0,MLR          ;CLEAR MAR.
14484          000001                      MICPC=MICPC+1
14485  034122  010000                      .WORD   .$.
14486  034124                              MOVE    # 0,MPR          ;CLEAR MAR.
14487          000002                      MICPC=MICPC+1
14488  034124  004000                      .WORD   .$.
14489  034126                              MOVE    BREG,OUT1 (CSR4)        ;CLEAR BSEL4.
14490          000003                      MICPC=MICPC+1
14491  034126  061224                      .WORD   .$.
14492  034130                              MOVE    BREG,OUT1 (CSR5)        ;CLEAR BSEL5.
14493          000004                      MICPC=MICPC+1
14494  034130  061225                      .WORD   .$.
14495  034132                              MOVE    BREG,OUT0 (0)           ;CLEAR IN DATA LB.
14496          000005                      MICPC=MICPC+1
14497  034132  062220                      .WORD   .$.
14498  034134                              MOVE    BREG,OUT0 (1)           ;CLEAR IN DATA HB.
14499          000006                      MICPC=MICPC+1
14500  034134  062221                      .WORD   .$.
14501  034136                              MOVE    BREG,OUT0 (4)           ;CLEAR IN BA LB.
14502          000007                      MICPC=MICPC+1
14503  034136  062224                      .WORD   .$.
14504  034140                              MOVE    BREG,OUT0 (5)           ;CLEAR IN BA HB.
14505          000010                      MICPC=MICPC+1
14506  034140  062225                      .WORD   .$.
14507  034142                              MOVE    BREG,SPAD (16)  ;RETURN ADDRESS PURPOSE.
```

```
14508                   000011
14509  034142  063236           MICPC=MICPC+1
14510  034144                   .WORD   .$.
14511                   000012  MOVE    # <HELDAT&377>,BREG     ;SET IN+BA LB.
14512  034144  000712           MICPC=MICPC+1
14513  034146                   .WORD   .$.
14514                   000013  MOVE    BREG,OUTO <4>           ;
14515  034146  062224           MICPC=MICPC+1
14516  034150                   .WORD   .$.
14517                   000014  MOVE    # <HELDAT/400&377>,BREG ;SET IN+BA+HB.
14518  034150  000470           MICPC=MICPC+1
14519  034152                   .WORD   .$.
14520                   000015  MOVE    BREG,OUTO <5>           ;
14521  034152  062225           MICPC=MICPC+1
14522  034154                   .WORD   .$.
14523                   000016  MOVE    # 022,MEM       ;
14524  034154  002422           MICPC=MICPC+1
14525  034156                   .WORD   .$.
14526                   000017  MOVE    MEM,OUT1 <CSR2> ;SET FOR HELL RAISER...
14527  034156  041222           MICPC=MICPC+1
14528  034160                   .WORD   .$.
14529                   000020  MOVE    # 357,BREG      ;GET THE MASK.
14530  034160  000757           MICPC=MICPC+1
14531  034162                   .WORD   .$.
14532                   000021  MOVE    INP1 <CSR11>,SPAD <0>   ;GET UPMS REGISTER.
14533  034162  123220           MICPC=MICPC+1
14534  034164                   .WORD   .$.
14535                   000022  AND     BREG,SPAD <0>,SPAD <0>  ;
14536  034164  063260           MICPC=MICPC+1
14537  034166                   .WORD   .$.
14538                   000023  MOVE    # 300,BREG              ;SET BR+RQ & VCTR:=XX4.
14539  034166  000700           MICPC=MICPC+1
14540  034170                   .WORD   .$.
14541                   000024  OR      BREG,SPAD <0>,OUT1 <11> ;
14542  034170  061311           MICPC=MICPC+1
14543  034172                   .WORD   .$.
14544                   000025  2$:     MOVE    INP1 <CSR11>,BREG       ;IS BR+RQ GRANTED???
14545  034172  120620           MICPC=MICPC+1
14546  034174                   .WORD   .$.
14547                   000026  BR7     2$                      ;NO,SPIN ON IT.
14548  034174  103425           MICPC=MICPC+1
14549  034176                   .WORD   .$.
14550                   000027  MOVE    # 0,BREG                ;
14551  034176  000400           MICPC=MICPC+1
14552  034200                   .WORD   .$.
14553                   000030  MOVE    BREG,OUT1 <CSR11>       ;CLEARS VCTR:XX4.
14554  034200  061231           MICPC=MICPC+1
14555  034202                   .WORD   .$.
14556                   000031  3$:     MOVE    INP1 <CSR2>,BREG        ;IS POP11 ALL SET???
14557  034202  120440           MICPC=MICPC+1
14558  034204                   .WORD   .$.
14559                   000032  BR4     3$                      ;NO,WAIT FOR IT.
14560  034204  103031           MICPC=MICPC+1
14561  034206                   .WORD   .$.
14562                   000033  4$:     MOVE    # 001,BREG              ;SET NPR+RQ BIT.
14563  034206  000401           MICPC=MICPC+1
                                .WORD   .$.
```

```
                                        MOVE    BREG,OUT1 (CSR10)       ;
                                        MICPC=MICPC+1
                                        .WORD   .$.
                        5$:             MOVE    INP1 (CSR10),BREG       ;IS NPR DONE?
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        BBO     6$
                                        MICPC=MICPC+1
                                        .WORD   .$.
                        3$:             MOVE    INP1 (CSR2),BREG        ;IS CSR LOADED???
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        BBI     4$                      ;NO WAIT FOR IT...
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    INP1 (CSR4),SPAD (1)    ;
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    INP1 (CSR5),SPAD (2)    ;
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        SBR     12$                     ;GO CHECK DATA...
                                        MICPC=MICPC+1
                                        .WORD   .$.
                        1:              MOVE    INP1 (CSR2),BREG        ;IS CSR LOADED???
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        BBI     5$                      ;NO,CHECK NPR DONE...
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    INP1 (CSR4),SPAD (1)    ;TRANSFER TO SPAD 1.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    INP1 (CSR5),SPAD (2)    ;TRANSFER TO SPAD 2.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        SBR     5$                      ;GO CHECK NPR.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                        12$:            MOVE    INPO (0),SPAD (0)       ;GET NPR DATA.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    SPAD (1),BREG           ;GET CSR DATA.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        $IFNE   BREG,SPAD (1)   15$     ;BR IF NO COMPARE.


                                        SUB2C   SPAD (1),BREG,NOP
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        BZ      64$
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        SBR     15$
                                        MICPC=MICPC+1
```

```
14620  034252  100462
14621  034254
14622  034254
14623          000056
14624  034254  023020
14625  034256
14626          000057
14627  034256  060602
14628  034260
14629
14630
14631  034260
14632          000060
14633  034260  060362
14634  034262
14635          000061
14636  034262  101472
14637  034264
14638          000062
14639  034264  061224
14640  034266
14641          000063
14642  034266  060600
14643  034270
14644          000064
14645  034270  061224
14646  034272
14647          000065
14648  034272  000425
14649  034274
14650          000066
14651  034274  061223
14652  034276
14653  034276
14654          000067
14655  034276  000471
14656  034300
14657          000070
14658  034300  104401
14659  034302
14660          000071
14661  034302  100400
14662  034304
14663  034304
14664          000072
14665  034304  000474
14666  034306
14667          000073
14668  034306  104454
14669  034310
14670          000074
14671  034310  100400
14672  034312  125125
14673  034314
14674  034314
14675
```

```
        .WORD   .$.
64$:
        MOVE    INPO (1),SPAD (0)       ;GET NPR DATA.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD (2),BREG           ;GET CSR DATA.
        MICPC=MICPC+1
        .WORD   .$.
        $IFEQ   BREG,SPAD (2)   16$     ;BRANCH IF GOOD.

        SUB2C   SPAD (2),BREG,NOP
        MICPC=MICPC+1
        .WORD   .$.
        BZ      16$
        MICPC=MICPC+1
        .WORD   .$.
15$:    MOVE    BREG,OUT1 (CSR4)        ;CSR DATA.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD (0),BREG           ;NPR DATA.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 (CSR4)        ;
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 25,BREG               ;ERROR TYPE.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 (CSR3)        ;
        MICPC=MICPC+1
        .WORD   .$.
        CALL    ERORI                   ;REPORT ERROR.
        MOVE    # (MICPC+3),BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     ERORI
        MICPC=MICPC+1
        .WORD   .$.
        SBR     15$                     ;LOOP ON ERROR.
        MICPC=MICPC+1
        .WORD   .$.
16$:    CALL    SCPE                    ;SCOPE THE TEST.
        MOVE    # (MICPC+3),BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     SCPE
        MICPC=MICPC+1
        .WORD   .$.
        SBR     64$                     ;YES DO NEXT PASS.
        MICPC=MICPC+1
        .WORD   .$.
HELDAT: 125125
$PWRI
$XZ
```

```
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,$TSTNM              ; LOAD THE NO. OF THIS TEST
14690  034322  012737  003662  001442          MOV     #SEOP,NEXT              ; POINT TO THE END OF PASS HANDLER.
14691                                                                  ;R1 CONTAINS BASE KMC11 ADDRESS
14692  034330  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
14693  034334  034350                          MCT65
14694  034336  104022                          ERROR   22                      ;TIME OUT ERROR...
14695  034340  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
14696  034344  000177  145072                  JMP     @NEXT                   ;GO TO NEXT TEST...
14697  034350                   MCT65:
14698  034350                   1$:     MOVE    # 0,BREG                       ;
14699          000000                   MICPC=MICPC+1
14700  034350  000400                   .WORD   .$.
14701  034352                           MOVE    BREG,SPAD <16>                ;FOR RETURN ADDRESS ...
14702          000001                   MICPC=MICPC+1
14703  034352  063236                   .WORD   .$.
14704  034354                           MOVE    BREG,SPAD <1>                 ;
14705          000002                   MICPC=MICPC+1
14706  034354  063221                   .WORD   .$.
14707  034356                           MOVE    BREG,SPAD <2>                 ;
14708          000003                   MICPC=MICPC+1
14709  034356  063222                   .WORD   .$.
14710  034360                           MOVE    BREG,SPAD <3>                 ;
14711          000004                   MICPC=MICPC+1
14712  034360  063223                   .WORD   .$.
14713  034362                           MOVE    # 220,BREG      ;GET READY TO SETUP POWER FAIL..
14714          000005                   MICPC=MICPC+1
14715  034362  000620                   .WORD   .$.
14716  034364                           MOVE    INP1 <CSR0>,SPAD <0>    ;GET CNTL/0 REGISTER..
14717          000006                   MICPC=MICPC+1
14718  034364  123000                   .WORD   .$.
14719  034366                           OR      BREG,SPAD <0>,OUT1 <CSR0>       ;SET RDO, POWER FAIL BIT..
14720          000007                   MICPC=MICPC+1
14721  034366  061300                   .WORD   .$.
14722  034370                           MOVE    # 200,BREG
14723          000010                   MICPC=MICPC+1
14724  034370  000600                   .WORD   .$.
14725  034372                           MOVE    INP1 <CSR11>,SPAD <0>   ;GET UPMS REGISTER.
14726          000011                   MICPC=MICPC+1
14727  034372  123220                   .WORD   .$.
14728  034374                           OR      BREG,SPAD <0>,OUT1 <CSR11>      ;SET BR+REQ,VCTR:=XX0..
14729          000012                   MICPC=MICPC+1
14730  034374  061311                   .WORD   .$.
14731  034376                   2$:     MOVE    INP1 <CSR11>,BREG       ;IS REQ GRANTED??
```

```
14732                   000013
14733  034376  120620           MICPC=MICPC+1
14734  034400                   .WORD   .$.
                                BR7     2$              ;NO, SPIN ON IT.
14735                   000014  MICPC=MICPC+1
14736  034400  103413           .WORD   .$.
14737  034402           3$:     MOVE    INP1 (CSR0),BREG        ;IS INTERRUPT SERVICED??
14738                   000015  MICPC=MICPC+1
14739  034402  120400           .WORD   .$.
14740  034404                   BR4     3$              ;NO, WAIT...
14741                   000016  MICPC=MICPC+1
14742  034404  103015           .WORD   .$.
14743                           ;:*
14744                           ;:*     NOW SET UP FOR FORCE POWER FAIL.
14745                           ;:*
14746  034406                   MOVE    # 002,BREG      ;GET READY TO SET FORCE POWER FAIL.
14747                   000017  MICPC=MICPC+1
14748  034406  000402           .WORD   .$.
14749  034410                   MOVE    INP1 (CSR11),SPAD (0)   ;GET UPMS REGISTER...
14750                   000020  MICPC=MICPC+1
14751  034410  123220           .WORD   .$.
14752  034412                   OR      BREG,SPAD (0),OUT1 (CSR11)      ;SET AC LO..
14753                   000021  MICPC=MICPC+1
14754  034412  061311           .WORD   .$.
14755  034414           4$:
14756
14757  034414                   MOVE    INP1 (CSR0),BREG        ;IS POWER FAIL SERVED??
14758                   000022  MICPC=MICPC+1
14759  034414  120400           .WORD   .$.
14760  034416                   BR7     6$              ;NO, UPDATE DELAY COUNT.
14761                   000023  MICPC=MICPC+1
14762  034416  103425           .WORD   .$.
14763  034420                   $BR     9$              ;YES, SCOPE THE TEST..
14764                   000024  MICPC=MICPC+1
14765  034420  100432           .WORD   .$.
14766  034422           6$:     $INC    SPAD (1)                ;UPDATE DELAY COUNT.
14767                   000025  MICPC=MICPC+1
14768  034422  063061   .WORD   .$.!.DSP
14769  034424                   $AOC    SPAD (2)                ;UPDATE DELAY COUNT.
14770                   000026  MICPC=MICPC+1
14771  034424  063102   .WORD   .$.!.DSP
14772  034426                   $AOC    SPAD (3)                ;UPDATE DELAY COUNT.
14773                   000027  MICPC=MICPC+1
14774  034426  063103   .WORD   .$.!.DSP
14775  034430                   BC      12$             ;IS IT DELAYED ENOUGH?
14776                   000030  MICPC=MICPC+1
14777  034430  101032           .WORD   .$.
14778  034432                   $BR     4$              ;NO, WAIT
14779                   000031  MICPC=MICPC+1
14780  034432  100422           .WORD   .$.
14781  034434           12$:
14782  034434           9$:
14783  034434                   MOVE    INP1 (CSR0),BREG        ;DUMMY WAIT LOOP
14784                   000032  MICPC=MICPC+1
14785  034434  120400           .WORD   .$.
14786  034436                   $BR     9$              ;DO THE NEXT PASS.
14787                   000033  MICPC=MICPC+1
```

```
 14788  034436  100432                      .WORD   .$.
 14789  034440                      TSTEN:

 14790
 14791
 14792                              ;SUBROUTINES
 14793                              ;-----------
 14794
 14795
 14796                              .EVEN
 14797                              ;*************************************************************************
 14798                              ;*
 14799                              ;*      ERROR REPORT ROUTINE. INTERRUPTS AT LOCATION XX0.
 14800                              ;*      1) CSR4:=XOR DATA       2) CSR5:=BAD DATA
 14801                              ;*      3) CSR3:=ERROR TYPE     4) CSR6:=ERROR PC IN MICRO-CODE.
 14802                              ;*      5) CSR7:=MISCALLANEOUS INFORMATION.
 14803                              ;*
 14804                              ;*************************************************************************
 14805                              ;
 14806  034440                              $BEGIN ,377
 14807  034440                              $LOC    .1000
 14808  034440                      ERORO:  $INC    SPAD (16)       ;PREPARE FOR RETURN.
 14809          000400                      MICPC=MICPC+1
 14810  034440  063076              .WORD   .$.!.DSP
 14811  034442                      ERORO1: $INC    SPAD (1F)
 14812          000401                      MICPC=MICPC+1
 14813  034442  063076              .WORD   .$.!.DSP
 14814  034444                              MOVE    BREG,OUT1 (CSR6)        ;ERROR PC.
 14815          000402                      MICPC=MICPC+1
 14816  034444  061226              .WORD   .$.
 14817  034446                              MOVE    BREG,SPAD (17)  ;SAVE RETURN ADDRESS.
 14818          000403                      MICPC=MICPC+1
 14819  034446  063237              .WORD   .$.
 14820  034450                              MOVE    # 201,BREG      ;SET UP RD+I & ERROR
 14821          000404                      MICPC=MICPC+1
 14822  034450  000601              .WORD   .$.
 14823  034452                              MOVE    INP1 (CSR0),SPAD (0)    ;GET THE BSEL0.
 14824          000405                      MICPC=MICPC+1
 14825  034452  123000              .WORD   .$.
 14826  034454                              OR      BREG,SPAD (0),OUT1 (CSR0)       ;SET RD+I & ERROR IN SEL0.
 14827          000406                      MICPC=MICPC+1
 14828  034454  061300              .WORD   .$.
 14829  034456                              MOVE    INP1 (CSR11),SPAD (0)   ;GET UPMS REGISTER.
 14830          000407                      MICPC=MICPC+1
 14831  034456  123220              .WORD   .$.
 14832  034460                              MOVE    # 357,BREG              ;GET THE MASK FOR PGMCLK.
 14833          000410                      MICPC=MICPC+1
 14834  034460  000757              .WORD   .$.
 14835  034462                              AND     BREG,SPAD (0),SPAD (0)  ;MASK OUT PGMCLK BIT.
 14836          000411                      MICPC=MICPC+1
 14837  034462  063260              .WORD   .$.
 14838  034464                              MOVE    # 200,BREG              ;PREPARE FOR INTERRUPT.
 14839          000412                      MICPC=MICPC+1
 14840  034464  000600              .WORD   .$.
 14841  034466                              OR      BREG,SPAD (0),OUT1 (CSR11)      ;SET BR+RO AT XX0.
 14842          000413                      MICPC=MICPC+1
 14843  034466  061311              .WORD   .$.
```

```
14844  034470
14845          000414
14846  034470  120620
14847  034472
14850          000415
14851  034472  107414
14852  034474
14853          000416
14854  034474  120400
14855  034476
14856          000417
14857  034476  106016
14860  034500
14861          000420
14862  034500  107423
14863  034502
14864          000421
14865  034502  060616
14866  034504
14867          000422
14870  034504  063017
14871  034506
14872          000423
14873  034506  000400
14874  034510
14875          000424
14876  034510  063236
14877  034512
14878          000425
14879  034512  061220
14880  034514
14881          000426
14882  034514  160617
14883
14884
14885
14886
14887
14888
14889
14890
14891
14892
14893
14894
14895
14896
14897
14898
14899
```

```
515:    MOVE    INP1 (CSR11),BREG       ;IS BR+RQ GRANTED?
        MICPC=MICPC+1
        .WORD   .$.
        BR7     515                     ;NO, SPIN ON IT
        MICPC=MICPC+1
        .WORD   .$.
525:    MOVE    INP1 (CSR0),BREG        ;IS ERROR INTERRUPT SERVICED?
        MICPC=MICPC+1
        .WORD   .$.
        BR0     525                     ;NO, SPIN ON IT.
        MICPC=MICPC+1
        .WORD   .$.
        BR7     525                     ;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   .$.
535:    MOVE    # 0,BREG                 ;
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,SPAD (16)  ;RESET SPAD (16)...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 (CSR0)        ;RESET THE TALK REGISTER...
        MICPC=MICPC+1
        .WORD   .$.
        $BR     SPAD (17) PAGE0 ;RETURN TO CALLER.
        MICPC=MICPC+1
        .WORD   .$.
;
;:*************************************************************************
;:*
;:*      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   .$.
        MOVE    # 202,BREG      ;BREG+RD+I, SCOP1.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 (CSR0)        ;SET RD+I, SCOP1.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    INP1 (CSR11),SPAD (0)   ;GET UPMS REGISTER
```

```
14900                   000434          MICPC=MICPC+1
14901   034530  123220                  .WORD   .$.
14902   034532                          MOVE    # 357,BREG              ;GET MASK FOR PGMCLK.
14903                   000435          MICPC=MICPC+1
14904   034532  000757                  .WORD   .$.
14905   034534                          AND     BREG,SPAD <0>,SPAD <0>  ;MASK OUT PGMCLK BIT.
14906                   000436          MICPC=MICPC+1
14907   034534  063260                  .WORD   .$.
14908   034536                          MOVE    # 200,BREG              ;PREPARE FOR INTERRUPT.
14909                   000437          MICPC=MICPC+1
14910   034536  000600                  .WORD   .$.
14911   034540                          OR      BREG,SPAD <0>,OUT1 <CSR11>      ;SET BR+RQ AT XXO.
14912                   000440          MICPC=MICPC+1
14913   034540  061311                  .WORD   .$.
14914   034542                  54$:    MOVE    INP1 <CSR11>,BREG       ;IS BR+RQ GRANTED?
14915                   000441          MICPC=MICPC+1
14916   034542  120620                  .WORD   .$.
14917   034544                          BR7     54$                     ;NO, SPIN ON IT.
14918                   000442          MICPC=MICPC+1
14919   034544  107441                  .WORD   .$.
14920   034546                  55$:    MOVE    INP1 <CSR0>,BREG        ;IS SCOP1 SERVICED?
14921                   000443          MICPC=MICPC+1
14922   034546  120400                  .WORD   .$.
14923   034550                          BR1     55$                     ;NO, SPIN ON IT.
14924                   000444          MICPC=MICPC+1
14925   034550  106443                  .WORD   .$.
14926   034552                          BR7     56$                     ;LOOP ON ERROR?
14927                   000445          MICPC=MICPC+1
14928   034552  107450                  .WORD   .$.
14929   034554                          MOVE    SPAD <16>,BREG  ;NO, POP THE RETURN ADDRESS.
14930                   000446          MICPC=MICPC+1
14931   034554  060616                  .WORD   .$.
14932   034556                          SADD    BREG,SPAD <17>,SPAD <17>        ;NO, POP THE RETURN ADDRESS.
14933                   000447          MICPC=MICPC+1
14934   034556  063017                  .WORD   .$.
14935   034560                  56$:    MOVE    # 0,BREG                ;NO, POP THE RETURN ADDRESS.
14936                   000450          MICPC=MICPC+1
14937   034560  000400                  .WORD   .$.
14938   034562                          MOVE    BREG,SPAD <16>  ;RESET SPAD <16>...
14939                   000451          MICPC=MICPC+1
14940   034562  063236                  .WORD   .$.
14941   034564                          MOVE    BREG,OUT1 <CSR0>        ;RESET THE TALK REGISTER.
14942                   000452          MICPC=MICPC+1
14943   034564  061220                  .WORD   .$.
14944   034566                          SBR     SPAD <17> PAGE0 ;RETURN TO CALLER.
14945                   000453          MICPC=MICPC+1
14946   034566  160617                  .WORD   .$.
14947
14948                           ;
14949                           ;:*****************************************************************************
14950                           ;:*
14951                           ;:*     SCOPE THE TEST ROUTINE.
14952                           ;:*
14953                           ;:*****************************************************************************
14954   034570                  SCPE0:  MOVE    BREG,SPAD <17>  ;SAVE RETURN ADDRESS.
14955                   000454          MICPC=MICPC+1
```

```
14956  034570  063237                .WORD   .$.
14957  034572                        MOVE    # 200,BREG      ;RD+0, XN BREG
14958          000455                MICPC=MICPC+1
14959  034572  000600                .WORD   .$.
14960  034574                        MOVE    BREG,OUT1 (CSR2)        ;BET RD+0 IN CSR2.
14961          000456                MICPC=MICPC+1
14962  034574  061222                .WORD   .$.
14963  034576                        MOVE    INP1 (CSR11),SPAD (0)   ;GET UPMS REG.
14964          000457                MICPC=MICPC+1
14965  034576  123220                .WORD   .$.
14966  034600                        MOVE    # 357,BREG              ;GET MASK FOR PGMCLK.
14967          000460                MICPC=MICPC+1
14968  034600  000757                .WORD   .$.
14969  034602                        AND     BREG,SPAD (0),SPAD (0)  ;MASK OUT PGMCLK BIT.
14970          000461                MICPC=MICPC+1
14971  034602  063260                .WORD   .$.
14972  034604                        MOVE    # 300,BREG      ;SET BR RQ & VCTR:=XX4.
14973          000462                MICPC=MICPC+1
14974  034604  000700                .WORD   .$.
14975  034606                        OR      BREG,SPAD (0),OUT1 (CSR11)      ;SET BR+RQ & VCTR:=XX4.
14976          000463                MICPC=MICPC+1
14977  034606  061311                .WORD   .$.
14978  034610                4$:     MOVE    INP1 (CSR11),BREG       ;IS BR+RQ GRANTED?
14979          000464                MICPC=MICPC+1
14980  034610  120620                .WORD   .$.
14981  034612                        BR7     4$              ;NO, SPIN ON IT.
14982          000465                MICPC=MICPC+1
14983  034612  107464                .WORD   .$.
14984  034614                        MOVE    # 00,BREG       ;PREPARE TO CLEAR VCTR:=XX4.
14985          000466                MICPC=MICPC+1
14986  034614  000400                .WORD   .$.
14987  034616                        MOVE    INP1 (CSR11),SPAD (0)   ;GET UPMS REG
14988          000467                MICPC=MICPC+1
14989  034616  123220                .WORD   .$.
14990  034620                        AND     BREG,SPAD (0),OUT1 (CSR11)      ;RESET XX4
14991          000470                MICPC=MICPC+1
14992  034620  061271                .WORD   .$.
14993  034622                5$:     MOVE    INP1 (CSR2),BREG        ;IS INTERRUPT SERVICED.
14994          000471                MICPC=MICPC+1
14995  034622  120440                .WORD   .$.
14996  034624                        BR7     5$              ;NO, SPIN ON IT.
14997          000472                MICPC=MICPC+1
14998  034624  107471                .WORD   .$.
14999  034626                        SBR     SPAD (17) PAGE0 ;RETURN
15000          000473                MICPC=MICPC+1
15001  034626  160617                .WORD   .$.
15002                        ;;
15003                        ;;*****************************************************************
15004                        ;;*
15005                        ;;*     MICRO-PROCESSOR PC SEQUENCE ERROR.
15006                        ;;*
15007                        ;;*****************************************************************
15008                        ;;
15009  034630                UPCSQR: MOVE    # 24,BREG                ;ERROR CODE.
15010          000474                MICPC=MICPC+1
15011  034630  000424                .WORD   .$.
```

```
15012  034632
15013                 000475
15014  034632  061223
15015  034634
15016                 000476
15017  034634  000601
15018  034636
15019                 000477
15020  034636  061220
15021  034640
15022                 000500
15023  034640  000600
15024  034642
15025                 000501
15026  034642  061231
15027  034644
15028                 000502
15029  034644  120620
15030  034646
15031                 000503
15032  034646  107502
15033  034650
15034                 000504
15035  034650  120400
15036  034652
15037                 000505
15038  034652  106104
15039  034654
15040  034654
15041                 000506
15042  034654  000510
15043  034656
15044                 000507
15045  034656  104454
15046  034660
15047                 000510
15048  034660  120400
15049  034662
15050                 000511
15051  034662  104510
15052  034664
15053
15054
15055
15056
15057
15058
15059
15060
15061
15062  034664
15063  034664  013737  001442  035356
15064  034672  013737  001206  035354
15065  034700  032711  000001
15066  034704  001012
15067  034706  032711  000002

                MOVE    BREG,OUT1 (CSR3)        ;SET ERROR TYPE.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 201,BREG              ;SET UP FOR ERROR REPORT.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT1 (CSR0)        ;
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 200,BREG              ;BREG+BR-RQ. AT VCTR:=XX0.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT1 (CSR11)       ;IS INTERRUPT SERVICED???
                MICPC=MICPC+1
                .WORD   .$.
1$:             MOVE    INP1 (CSR11),BREG       ;IS INTERRUPT SERVICED??
                MICPC=MICPC+1
                .WORD   .$.
                BR7     1$                      ;SPIN ON IT.
                MICPC=MICPC+1
                .WORD   .$.
2$:             MOVE    INP1 (CSR0),BREG        ;IS ERROR SERVICED???
                MICPC=MICPC+1
                .WORD   .$.
                BR0     2$                      ;WAIT...
                MICPC=MICPC+1
                .WORD   .$.
                CALL    SCPED                   ;SCOPE THE FAILING TEST.
                MOVE    # (MICPC+3),BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     SCPED
                MICPC=MICPC+1
                .WORD   .$.
3$:             MOVE    INP1 (CSR0),BREG        ;WAIT FOR PDP11 TO
                MICPC=MICPC+1
                .WORD   .$.
                SBR     3$                      ;.....................
                MICPC=MICPC+1
                .WORD   .$.
MCSREN:
;;
;;*****************************************************
;;*
;;*      INPUT INTERRUPT SERVICE ROUTINE
;;*      1. TAKES CARE OF ERROR REPORTS.
;;*      2. PROVIDE SERVICE FOR SCOPE DATA.
;;*
;;*****************************************************
;;
INISR:  ;MOV    R0,-(SP)        ;PUSH R0.
        MOV     NEXT,SVNXT      ;SAVE NEXT TEST ADDRESS.
        MOV     SLPADR,SVLPDR   ;SAVE LOOP ADDRESS.
        BIT     #BIT0,(R1)      ;IS ERROR BIT SET?
        BNE     ERR1            ;YES, SERVE THE ERROR INTERRUPT.
        BIT     #BIT1,(R1)      ;IS IT SCOPE DATA SET?
```

```
                                                        BNE     SCPDT           ;YES, GO SERVE IT.
                    000020                              BIT     #BIT4,(R1)      ;IS PWR. FAIL SERVICE SET?
                                                        BNE     PWFAIL          ;GO SERVE POWER FAIL INTERRUPT.
                                                        ;
                    011032                              TYPE    ,MILINT         ;TYPE ILLEGAL INTERRUPT.
                                                        HALT                    ;WAIT FOR OPERATOR ACTION.
                                                        BR      RETN1           ;CONTINUE....
                    035022  001442          ERR1:       MOV     #3$,NEXT        ;LOAD RETURN ADDRESS.
                    001206                              MOV     #3$,SLPADR      ;LOAD RETURN ADDRESS.
                    000004  001224                      MOVB    4(R1),SCNMOT    ;LOAD GOOD DATA.
                    000005  001226                      MOVB    5(R1),          ;LOAD BAD DATA.
                    000006  001222                      MOVB    6(R1),          ;LOAD UP PC.
                    000377  035020                      BIC     #377,           ;CLEAR THE ERROR TYPE FIELD.
                    000003  035020                      BISB    3(R1),6$        ;SET THAT IN ERROR TRAP.
                    000007  001220                      MOVB    7(R1),SCSRCR    ;SET CSR FOR REFERENCE.
                    035354  007070                      MOV     #SVLPDR,ERTAB1  ;SAVE ERROR PC...
                                                        ;
                    104000                      6$:     ERROR                   ;REPORT THE ERROR.
                    001460  007070              3$:     MOV     #SAVPC,ERTAB1   ;RESTORE ERROR PC POINTER.
                    000400  144202                      BIT     #SW08,@SWR      ;GO TO TOP OF TEST?
                    001020                              BNE     1$              ;BRANCH IF YES.
                    002000  144172                      BIT     #SW10,@SWR      ;GO TO NEXT TEST?
                    001412                              BEQ     2$              ;BRANCH IF NO.
                    035356  001206                      MOV     SVNXT,SLPADR    ;SET FOR NEXT TEST.
                    001200                              MOV     #STACK,SP       ;RESET SP.
                    000200  177776                      MOV     #PR4,PS         ;SET PRIORITY 4 SO KMC CAN INTRRUPT.
                    000177  144112                      JMP     @SLPADR         ;GO TO SPECIFIED TEST.
                    000200                      2$:     BIC     #BIT7,(R1)      ;CLEAR LOOP ON ERROR BIT.
                                                1$:     ;MOV    #1,CLKCTR       ;RESET CLOCK.
                    000001                              BIC     #BIT0,(R1)      ;CLEAR ERROR BIT IN BSELO.
                    000511                              BR      RETN1           ;RETURN.
                    001444          SCPDT:      CLR     LOCK                    ;CLEAR FLAG
                                                SCOP1                           ;SCOPE THE DATA.
                    001444                      TST     LOCK                    ;IS FLAG SET?-I.E SCOPE THE DATA.
                    001002                      BNE     SCPDT1                  ;YES, GO SET ACCORDINGLY.
                    000200                      BIC     #BIT7,(R1)              ;CLEAR SCOPE DATA LOOP BIT.
                                                SCPDT1:
                    000002                      BIC     #BIT1,(R1)              ;CLEAR SCOPE DATA BIT.
                    000476                      BR      RETN1                   ;RETURN.
                                                ;:*     THIS IS PART OF POWER FAIL TEST
                                                ;:*     AFTER KMC CAUSES FORCE POWER FAIL
                                                ;:*     AND SETTING UP FOR POWER FAIL THIS
                                                ;:*     FUNCTIONS DONE HERE.
                    035202  001442          PWFAIL: MOV     #3O$,NEXT       ;RESTORE.
                    005037  011112                  CLR     TEMP
                    035202  001206                  MOV     #3O$,SLPADR     ;RESTORE.
                    013746  000024                  MOV     @#24,-(SP)      ;STORE POWER FAIL ADDRESS.
                    012737  035204  000024          MOV     #12$,@#24       ;SET UP FOR FORCE POWER FAIL.
                    042711  000020                  BIC     #BIT4,(R1)      ;SIGNAL POWER FAIL SET UP.
                    005237  011112          15$:    INC     TEMP            ;WAIT FOR POWER FAIL.
                    001375                          BNE     15$             ;BR IF DELAY NOT DONE.
                    104017                          ERROR   17              ;ERROR, NO POWER FAIL.
                    000427                  3O$:    BR      33$             ;
                                                    ;:*
                                                    ;:*     RESTARTS HERE IN CASE OF POWER FAIL.
                                                    ;:*
```

```
 15124  035204  012737  035222  000024  12$:    MOV     #21$,@#24       ;POWER UP ADDRESS.
 15125  035212  010637  035220          21$:    MOV     SP,24$          ;STORE STACK.
 15126  035216  000000                  24$:    HALT                    ;WAIT FOR POWER UP SEQUENCE.
 15127  035220  000000                          .WORD   0               ;
 15128  035222  013706  035220          21$:    MOV     24$,SP          ;RESTORE STACK.
 15129  035226  062706  000004                  POP2SP                  ;POP STACK TWICE.
 15130  035232  012737  035262  001206          MOV     #33$,SLPADR     ;
 15131  035240  012737  035262  001442          MOV     #33$,NEXT       ;SAVE FOR RETURN ON ERROR.
 15132  035246  010637  000024                  MOV     (SP)+,@#24      ;RESTORE TRUE POWER FAIL ADDRESS.
 15133  035252  022737  007100  000024          CMP     @#PWRVEC,#24    ;IS IT CORRECT?
 15134  035260  001401                          BEQ     33$             ;BR IF YES.
 15135  035262  104017                          ERROR   17              ;ERROR, STACK IS IN CORRECT.
 15136  035264  012737  007100  000024  33$:    MOV     #PWRVEC,@#24    ;RESTORE TRUE POWER FAIL ADDRESS.
 15137  035272  016706  001200                  MOV     #STACK,SP       ;RESET STACK...
 15138  035276  005711                          TST     (R1)            ;IS KMC STILL RUNNING.E.E.INIT BLOCKED.
 15139  035300  100407                          BMI     RETN2           ;YES RETURN.
 15140  035302  012737  035316  001206          MOV     #RETN2,SLPADR
 15141  035310  012737  035316  001442          MOV     #RETN2,NEXT
 15142  035316  104016                          ERROR   16              ;ERROR, BUS INIT WAS NOT BLOCKED.
 15143  035320  013737  035356  001206  RETN2:  MOV     SVNXT,SLPADR    ;GO TO NEXT TEST.
 15144  035326  000177  143656                  JMP     @SLPADR         ;
 15145  035332  012737  000001  036150  RETN1:  MOV     #1,CLKCTR       ;RESET CLOCK.
 15146  035340  013737  035356  001442          MOV     SVNXT,NEXT      ;RESTORE:::
 15147  035346  013737  035354  001206          MOV     SVLPDR,SLPADR   ;RESTORE...
 15148  035354  000002                          RTI
 15149  035356  000000          SVLPDR: 0
 15150  035360  000000          SVNXT:  0
 15151
 15152
 15153
 15154                                          ;
 15155                                          ;*****************************************************
 15156                                          ;*
 15157                                          ;*      OUTPUT INTERRUPT SERVICE ROUTINE
 15158                                          ;*      1. TAKE CARE OF SCOPE TEST
 15159                                          ;*
 15160                                          ;*****************************************************
 15161                                          ;
 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,SVLPAD   ;SAVE LOOP ADDRESS.
 15165  035376  012737  035452  001206          MOV     #SKOPE,SLPADR   ;STORE RETURN ADDRESS.
 15166  035404  000004                          SCOPE                   ;SCOPE THE TEST.
 15167  035406  013737  035534  001206          MOV     SVLPAD,SLPADR   ;NO ITERATION.
 15168  035414  005137  036146                  COM     DONE            ;SETUP DONE FLAG
 15169  035420  000432                          BR      RETI
 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                  CLRB    2(R1)           ;CLEAR CSR2.
 15174  035450  000416                          BR      RETI
 15175  035452  013737  035534  001206  SKOPE:  MOV     SVLPAD,SLPADR   ;RESTORE RETURN ADDRESS.
 15176                                  ;       BIC     #BIT7,2(R1)     ;CLEAR RO=0.
 15177  035460  122761  000024  000003          CMPB    #24,3(R1)       ;IS IT UPC SQ. ERROR???
 15178  035466  001013                          BNE     RET2            ;NO,THEN LET IT BE SCOPED BY
 15179                                                                  ;MICRO-PROCESSOR SCOPE ROUTINE.
```

```
15180  035470  105061  000003                  CLRB    3(R1)           ;CLEAN THE WORLD...
15181  035474  013737  001206  001442          MOV     $LPADR,NEXT     ;ELSE, START FROM BEGINING.
15182  035502  005137  036146                  COM     DONE            ;
15183  035506  012737  000001  036150  RET1:   MOV     #1,CLKCTR       ;RESET CLOCK.
15184  035514  000002                          RTI                     ;RETURN TO WAIT CLOCK.
15185  035516  012737  000001  036150  RET2:   MOV     #1,CLKCTR               ;RESET WAIT CLOCK.
15186  035524  142761  000200  000002          BICB    #BIT7,2(R1)     ;CLEAR RD+0.
15187  035532  000002                          RTI                     ;RETURN.
15188  035534  000000                  SVLPAD: 0
15189
15190
15191
15192
15193                                   ;
15194                                   ;:*****************************************************************
15195                                   ;:*
15196                                   ;:*     ROUTINE TO LOAD MICRO-CODE INTO CRAM...
15197                                   ;:*     1. LOADS TEST.
15198                                   ;:*     2. LOADS UTILITY ROUTINES...
15199                                   ;:*     R1 CONTAINS THE CSR AT THE TIME OF ENTRY...
15200                                   ;:*
15201                                   ;:*****************************************************************
15202
15203                                   ;:*************************************
15204  035536                           ;
15205  035536                           LDVRUT:
15206  035536                           MCRAM:  ;CLR    R0                      ;R0=POINTS TO CRAM ADDRESS.
15207  035536  012700  002000                   MOV     #2000,R0                ;
15208  035542  012711  040000                   MOV     #BIT14,(R1)             ;MASTER CLEAR KMC11
15209  035546  042711  140000                   BIC     #BIT15!BIT14,(R1)       ;AND SHUT IT OFF.
15210  035552  012702  104474                   MOV     #104474,R2              ;R2=ADR UPCSR(MICRO-INSTRUCTION)
15211  035556  005011                   6$:     CLR     (R1)                    ;START WITH CLEAN WORLD.
15212  035560  010061  000004                   MOV     R0,4(R1)                ;LOAD CRAM ADDRESS.
15213  035564  010261  000006                   MOV     R2,6(R1)                ;LOAD INSTRUCTION.
15214  035570  012711  002000                   MOV     #BIT10,(R1)             ;SET RUN0.
15215  035574  012711  022000                   MOV     #BIT13!BIT10,(R1)       ;WRITE IT.
15216  035600  005300                           DEC     R0                      ;COUNT BY ONE.
15217  035602  001365                           BNE     6$                      ;BR IF NOT DONE.
       035604  013602                           MOV     @(SP)+,R2               ;
       035606  036152                           MOV     R2,TSTRT                ;SAVE START ADDRESS...
       035612  000002                           ADD     #2,-(SP)                ;       ;POP RETURN ADDRESS...
       035614  013737  001442  036154           MOV     NEXT,MCEND              ;SET UP END POINTER.
       035622  022737  003662  036154           CMP     #SEOP,MCEND             ;IS IT LAST TEST?
       035630  001003                           BNE     1$                      ;NO, THEN IT'S O.K.
       035632  012737  034440  036154           MOV     #TSTEN,MCEND            ;SET UP END POINTER.
       035640  005011                   1$:     CLR     (R1)                    ;START WITH CLEAN WORLD.
       035642  010061  000004                   MOV     R0,4(R1)                ;LOAD CRAM ADDRESS...
       035646  012261  000006                   MOV     (R2)+,6(R1)             ;LOAD WORD TO BE WRITTEN.
       035652  012711  002000                   MOV     #BIT10,(R1)             ;WRITE IT...
       035656  012711  022000                   MOV     #BIT13!BIT10,(R1)       ;WRITE IT.
       035662  005200                           INC     R0                      ;UPDATE CRAM ADDRESS.
       035664  022700  002000                   CMP     #2000,R0                ;OVER FLOW??
15231  035670  002403                           BLT     2$                      ;YES, THATS IT! NO MORE.
15232  035672  023702  036154                   CMP     MCEND,R2                ;IS IT DONE?
15233  035676  003360                           BGT     1$                      ;NO,CONTINUE LOADING.
15234  035700  022737  034664  036154   2$:     CMP     #MCSREN,MCEND           ;IS UTILITY ALSO LOADED?
15235  035706  001410                           BEQ     3$                      ;YES,RETURN.
```

```
 15236  035712  012700  000400              MOV     #400,R0         ;R0=POINTS CRAM ADDRESS.
 15237  035716  012702  034440              MOV     #ERORO,R2       ;R2:=POINTS TO MICRO-CODE.
 15238  035722  012737  034664  036154      MOV     #MCSREN,MCEND   ;SET UP END POINTER.
 15239  035730  000744                      BR      1$              ;START LOADING UTILITY.
 15240  035732  005011              3$:     CLR     (R1)            ;END WITH CLEAN WORLD.
                                ;
                                ;;*****************************************************************
                                ;;*
                                ;;*     VERIFY THE MICRO-CODE LOADED.
                                ;;*     VERIFIES THE LOADED TEST MICRO-CODE. AND UTILITY
                                ;;*     ROUTINES...
                                ;;*     R1 CONTAINS DEVICE CSR AT THE TIME OF ENTRY...
                                ;;*
                                ;;*****************************************************************
                                ;
                                ;;************************************
                                ;
 15241  035734  005002          VERFY:  CLR     R2              ;R2 GETS CRAM ADDRESS.
 15242  035736  013700  036152          MOV     TSTRT,R0        ;LOAD START ADDRESS...
 15243  035742  013737  001442  036154  MOV     NEXT,MCEND      ;SET UP END POINTER.
 15244  035750  022737  003662  036154  CMP     #SEOP,MCEND     ;IS IT LAST TEST??
 15245  035756  001003                  BNE     1$              ;NO,THEN IT'S O.K.
 15246  035760  012737  034440  036154  MOV     #TSTEN,MCEND    ;SET UP END POINTER.
 15247  035766  005011          1$:     CLR     (R1)            ;CLEAR SELD.
 15250  035770  010261  000004          MOV     R2,4(R1)        ;SET THE CRAM ADDRESS.
 15251  035774  012711  002000          MOV     #BIT10,(R1)     ;SET ROMD.
 15252  036000  026120  000006          CMP     6(R1),(R0)+     ;CHECK IF RIGHT?
 15253  036004  001403                  BEQ     3$              ;CHECK IF DONE?
 15254  036006  104401  036156          TYPE    ,MLDER          ;LOADING ERROR.
 15255  036012  000000                  HALT                    ;WAIT FOR OPERATOR.
 15256  036014  005202          3$:     INC     R2              ;BUMP CRAM ADDRESS.
 15257  036016  023700  036154          CMP     MCEND,R0        ;IS IT DONE??
 15260  036022  003361                  BGT     1$              ;NO,CHECK NEXT.
 15261  036024  022737  034664  036154  CMP     #MCSREN,MCEND   ;IS IT DONE???
 15262  036032  001410                  BEQ     6$              ;YES,RETURN...
 15263  036034  012702  000400          MOV     #400,R2         ;R2 CONTAINS CRAM ADDRESS...
 15264  036040  012737  034664  036154  MOV     #MCSREN,MCEND   ;SET UP END POINTER...
 15265  036046  012700  034440          MOV     #ERORO,R0       ;R0=POINTS TO MICRO-CODE.
 15266  036052  000745                  BR      1$              ;START CHECKING UTILITIES.
 15267  036054  005011          6$:     CLR     (R1)            ;END WITH CLEAN WORLD.
 15270  036056  012711  040000          MOV     #BIT14,(R1)     ;MASTER CLEAR THE KMC.
 15271  036062  042711  140000          BIC     #BIT14!BIT15,(R1)
                                ;
                                ;;*****************************************************************
                                ;;*
                                ;;*     WAIT CLOCK :- WAITS FOR KMC TO RESPOND
                                ;;*     BACK WITH DONE OR ERROR IN SPECIFIC TIME IF DOES NOT
                                ;;*     GET ONE GENERATES A TIME OUT ERROR.
                                ;;*
                                ;;*****************************************************************
                                ;
 15272  036066  005037  036150  WATCLK: CLR     CLKCTR          ;INITIALIZE CLOCK.
 15273  036072  005037  036146          CLR     DONE            ;WAIT.
```

```
                                                MOV     #BIT15,(R1)     ;INITIATE THE TEST...
                        012711  100000          1$:     TST     DONE    ;IS DONE FLAG SET?
                        005737  036146          BNE     3$              ;YES, START THE NEXT TEST.
                        001010                  INC     CLKCTR          ;COUNT BY ONE.
                        005237  036150          DEC     CLKCTR          ;WAIT CLOCK.
                        005337  036150          INC     CLKCTR          ;  "    "
                        005237  036150          BNE     1$              ;GO CHECK AGAIN. IF NOT TIMED OUT.
                        001366                  RTS     PC              ;
                        000207
                        013737  001442  001206  3$:     MOV     NEXT,SLPADR     ;SET THE NEXT TEST ADDRESS.
                        012706  001200          MOV     #STACK,SP       ;POP THE STACK.
                        000177  143040          JMP     @SLPADR         ;START THE NEXT TEST.
                                                DONE:   0
                                                CLKCTR: 0
                                                TSTRT:  0
                                                MCEND:  0
                        046200  040517  044504  MLDER:  .ASCIZ  <200>/LOADING ERROR/
                        043516  042440  051122
                        051117  000
                        036176                  .EVEN
                                                $EINI
                        044600  052502  027523  EM1:    .ASCIZ  <200>↑IBUS/IBUS# REGISTER DATA TEST↑
                            200  041111  051523  EM2:    .ASCIZ  <200>↑IBUS/IBUS# REGISTER DUAL ADDRESSING TEST↑
                            200  051102  051040  EM3:    .ASCIZ  <200>/MR REGISTER DATA TEST/
                        051600  051103  052101  EMH:    .ASCIZ  <200>/SCRATCH PAD DATA TEST/
                            200  041523  040522  EM5:    .ASCIZ  <200>/SCRATCH PAD DUAL ADDRESSING TEST/
                            200  040515  047111  EM6:    .ASCIZ  <200>/MAIN MEMORY DATA TEST/
                        046600  044501  020116  EM7:    .ASCIZ  <200>/MAIN MEMORY DUAL ADDRESSING TEST/
                        040600  052125  020117  EM10:   .ASCIZ  <200>/AUTO MAINC FUNCTION TEST/
                            200  050116  020122  EM11:   .ASCIZ  <200>/MPR TEST/
                            200  052515  052114  EM12:   .ASCIZ  <200>/MULTIPLE MPR TEST/
                        047200  047117  042440  EM13:   .ASCIZ  <200>/NON EX MEM FAILED TO SET/
                        050200  047522  051107  EM14:   .ASCIZ  <200>/PROGRAM CLOCK TEST/
                        040600  052514  043040  EM15:   .ASCIZ  <200>/ALU FUNCTION WITH C BIT CLEAR TEST/
                        050200  052517  051105  EM16:   .ASCIZ  <200>/POWER FAIL; BUS INIT WAS NOT BLOCKED/
                        043600  051117  042503  EM17:   .ASCIZ  <200>/FORCE POWER FAIL ERROR/
                        047200  044517  042523  EM20:   .ASCIZ  <200>/NOISE TEST ON IBUS#,IBUS,SPAD,MEMORY/
                        040600  052514  041440  EM21:   .ASCIZ  <200>/ALU C BIT TEST FAILURE/
                        052200  046511  020105  EM22:   .ASCIZ  <200>/TIME OUT ERROR/
                        040600  052514  043040  EM23:   .ASCIZ  <200>/ALU FUNCTION TEST WITH C BIT SET/
                        052600  041520  051440  EM24:   .ASCIZ  <200>/UPC SEQUENCE ERROR/
                        044200  046105  020114  EM25:   .ASCIZ  <200>/HELL RAISER TEST/

                        043600  047517  020104  DH1:    .ASCIZ  <200>/GOOD     BAD    UPC    REGISTER/
                            200  047507  042117  DH2:    .ASCIZ  <200>/GOOD     BAD/
                        043600  047517  020104  DH3:    .ASCIZ  <200>/GOOD     BAD    UPC    ADDRESS/
                            200  047507  042117  DHH:    .ASCIZ  <200>/GOOD     BAD    UPC    FUNCTION/
                        037434                  .EVEN

                        000004                  DT1:    4
                            003      006                .BYTE   3,6
                        001224                          $GDOAT
                            003      004                .BYTE   3,4
                        001226                          $BDOAT
                            003      006                .BYTE   3,6
                        001222                          $BDAOR
                            003      006                .BYTE   3,6
```

```
037454  001220                          $GDADR
037456  000002          DT2:            2
037460     003  004                     .BYTE   3,4
037462  001224                          $GDDAT
037464     003  004                     .BYTE   3,4
037466  001226                          $SDDAT
037470          CORMAX:
        000001                          .END
```

```
ABASE = 000000            298     339
ACOM1 = 000000            298     341
ACOM2 = 000000            298     342
ACPUOP= 000000            298     313
ADDM0 = 000000            298     343
ADDM1 = 000000            298     344
ADDM10= 000000            298     353
ADDM11= 000000            298     354
ADDM12= 000000            298     355
ADDM13= 000000            298     356
ADDM14= 000000            298     357
ADDM15= 000000            298     358
ADDR2 = 000000            298     345
ADDR3 = 000000            298     346
ADDR4 = 000000            298     347
ADDR5 = 000000            298     348
ADDR6 = 000000            298     349
ADDR7 = 000000            298     350
ADDR8 = 000000            298     351
ADDR9 = 000000            298     352
ADEVCT= 000000            298     304
ADEVM = 000000            298     340
ADRCNT  006047           1400*   1415*   1424#
ADVANC= 104420           1569#
AENV  = 000002              1#     298      309
AENVM = 000000            298     310
AFATAL= 000000            298     301
ANNOR1= 000000            298     326
ANNOR2= 000000            298     330
ANNOR3= 000000            298     333
ANNOR4= 000000            298     336
ANNRS1= 000000            298     320
ANNRS2= 000000            298     323
ANNRS3= 000000            298     331
ANNRS4= 000000            298     334
ANSDCP= 000000            298     306
ANSELC= 000000            298     307
ANSTY = 000000            298     300
ANTYP1= 000000            298     321
ANTYP2= 000000            298     324
ANTYP3= 000000            298     332
ANTYP4= 000000            298     335
APASS = 000000            298     303
APRIOR= 000000            298
APTCSU= 000040           1125    1230#
APTENV= 000001           1118    1186    1228#    1630
APTSIZ= 000200           1227#
APTSPO= 000100           1120    1188    1229#
APT.SI  013552            796    2210#
ASWREG= 000000            298     311
ATESTM= 000000            298     302
AUDONE  003354            833     854     893#
AUNIT = 000000            298     305
AUSTRT  003126            832#
AUSWR = 000000            298     312
AUTO.S  012152            794    1954#
```

```
AVECT1= 000000  298   337
AVECT2= 000000  298   338
BASE    013770 2221* 2232  2246* 2256#
BINBND  006402 1472* 1475* 1476  1513#
BIT0    = 000001  161#  1737  1738  15065  15097
BIT00   = 000001  151#  161
BIT01   = 000002  150#  160
BIT02   = 000004  149#  159
BIT03   = 000010  148#  158
BIT04   = 000020  147#  157
BIT05   = 000040  146#  156
BIT06   = 000100  145#  155
BIT07   = 000200  144#  154
BIT08   = 000400  143#  153
BIT09   = 001000  142#  152
BIT1    = 000002  160#  801  1731  1737  1738  2091  15067  15105  15172
BIT10   = 002000  141#  2072  15213  15214  15227  15228  15263
BIT11   = 004000  140#  1064
BIT12   = 010000  139#  2026  2089
BIT13   = 020000  138#  2029  2075  2093  15214  15228
BIT14   = 040000  137#  1046  2040  2042  2093  2104  15207  15208  15278  15279
BIT15   = 100000  136#  741  15208  15279  15292
BIT2    = 000004  159#  801  942  1738
BIT3    = 000010  158#  2095  2102
BIT4    = 000020  157#  1721  1744  1746  15069  15116  15162  15170
BIT5    = 000040  156#  2207
BIT6    = 000100  155#  1726  1727  2097
BIT7    = 000200  154#  1727  1846  2207  15095  15103  15186
BIT8    = 000400  153#  2084  2086  2099  2101  2107  2110  2167
BIT9    = 001000  152#  2081  2107  2110  2165  2167
BM      010603 1769#  2053
BPTVEC= 000014  168#
BRCR    = 000011  1#
BRLVL   013504 2162  2172  2180  2193#
BRW     004354  947  1080#
CHRCNT  006400 1470* 1473  1477  1493* 1511#  1512
CKSWR   011216  772  1086  1576  1794#
CKSWR1  011272 1806# 1818
CKSWR2  011304 1809#
CKSWR3  011310 1811#
CKSWR4  011314 1812# 1820  1827
CKSWR5  011420 1795  1797  1801  1836#
CLKCTR  036150 15145* 15183* 15185* 15290* 15295* 15296* 15297* 15304#
CLKX    001452  373#
CMPRE   034034 14057 14067 14077 14087 14097 14107 14117 14129 14139 14149 14159 14169 14179
               14189 14199 14215 14234 14247 14260 14273 14286 14299 14312 14325 14338 14416#
CNERR   011001  866  1769#
CNT.MA  002302  387  621#  740  742  744  1874
CNVRT   = 104417  863  972  974  976  978  1568#  1611  1613  1688  1809
COMERR  010734  861 1769#
COMM    010643 1769# 2031
CONTAB  003330  870  878#
CONVRT= 104416  813  869  1567#  1627  1976
CORMAX  037470 15312#
CR      = 000015  76#  1164  1174
CREAM   001502  386#  739*  1870*  1871  1873*  1877
```

```
CRLF   = 000200          77#  1135  1174
CSR      010335        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 14556 14557 14574 14580 14961 14994
CSR3   = 000003           1#  2321  2401  2510  2598  2715  2795  2912  2992  3119  3208  3345  3443
                        3560  3640  3757  3837  3964  4034  4151  4231  4348  4428  4545  4625  4742
                        4822  4939  5019  5133  5203  5319  5404  5526  5611  5733  5818  5940  6025
                        6147  6232  6354  6438  6561  6646  6768  6853  6975  7060  7182  7267  7437
                        7471  7674  7714  7882  8072  8191  8348  8501  8591  8751  8878  9193  9510
                        9827  10144 10461 10778 11095 11412 11729 12046 12363 12680 12997 13314 13631
                       13738  13770 13880 13882 14063 14329 14444 14560 15013
CSR4   = 000004           1#  2312  2392  2509  2589  2706  2786  2903  2983  3110  3199  3336  3434
                        3551  3631  3748  3828  3945  4025  4142  4222  4339  4419  4536  4616  4733
                        4813  4930  5010  5127  5147  5307  5382  5514  5594  5721  5806  5928  6013
                        6135  6220  6342  6427  6549  6634  6756  6841  6963  7048  7170  7255  7431
                        7465  7682  7702  8060  8182  8339  8496  8585  8745  9184  9501  9818  10135
                       10452  10769 11086 11403 11720 12037 12354 12671 12988 13305 13622 13885 14079
                       14367  14438 14460 14580 14598 14656 14744
CSR5   = 000005           1#  2315  2395  2512  2592  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  5388  5520  5605  5727  5812  5934  6019
                        6141  6224  6346  6433  6555  6640  6762  6847  6969  7054  7176  7261  7425
                        7462  7668  7708  8066  8185  8342  8499  8579  8739  9187  9504  9821  10138
                       10455  10772 11089 11406 11723 12040 12357 12674 12991 13308 13625 13888 14089
                       14370  14432 14453 14583 14598
CSR6   = 000006           1#  13882 14059 14615
CSR7   = 000007           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  5325  5410  5532  5617  5739  5824  5946  6031  6153  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  1625  1628#
DATACL=  104413        1564#
DATAHD   006742        1599#  1621  1624#
DATIN  = 000001           1#
DATIO  = 000000           1#
DATI1  = 000001           1#
DATOUT= 000021            1#
DATOO  = 000002           1#
DATO1  = 000003           1#
DDISP  = 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#
```

```
DH3     037334          433     436     439   15312#
DH4     037373          454     472   15312#
DISPLA  001242          268#    758#    764#   1076#
DISPRE  000174          229#    764
DONE    036146        15168#  15182#  15291#  15293   15303#
DSWR  = 177570           82#    238#    267
DT1     037434          419     422     428     431     434     437     440     455     473   15312#
DT2     037456          425     443     446     464     479   15312#
DZDME = ****** U        414
DZDMG = ****** U        414
DZKCA = 000000         1078    1089    1903    2275    2472    2669    2866    3064    3281    3514    3711    3908    4105
                       4302    4499    4696    4893    5091    5268    5475    5682    5889    6096    6303    6510    6717
                       6924    7131    7337    7511    7754    7891    8131    8288    8449    8662    8828    8954    9271
                       9588    9905   10222   10539   10856   11173   11490   11807   12124   12441   12758   13075   13392
                      13707   13850   14474   14692
EMTVEC= 000030          171#
EM1     036176          417   15312#
EM10    036520          438   15312#
EM11    036553          441   15312#
EM12    036565          444   15312#
EM13    036610          447   15312#
EM14    036642          450   15312#
EM15    036666          453   15312#
EM16    036732          456   15312#
EM17    037000          459   15312#
EM2     036235          420   15312#
EM20    037030          462   15312#
EM21    037076          465   15312#
EM22    037126          468   15312#
EM23    037146          471   15312#
EM24    037210          474   15312#
EM25    037234          477   15312#
EM3     036307          423   15312#
EM4     036336          426   15312#
EM5     036365          429   15312#
EM6     036427          432   15312#
EM7     036456          435   15312#
ERCT00  002304          623#
ERCT01  002310          626#
ERCT02  002314          629#
ERCT03  002320          632#
ERCT04  002324          635#
ERCT05  002330          638#
ERCT06  002334          641#
ERCT07  002340          644#
ERCT10  002344          647#
ERCT11  002350          650#
ERCT12  002354          653#
ERCT13  002360          656#
ERCT14  002364          659#
ERCT15  002370          662#
ERCT16  002374          665#
ERCT17  002400          668#
EROR  = 035350       2281#   2335    2415   2478#   2532    2612   2675#   2729    2809   2872#   2926    3006   3070#
                     3133    3222   3287#   3359    3457   3520#   3574    3654   3717#   3771    3851   3914#   3968
                     4048   4111#    4165    4245   4308#   4362    4442   4505#   4559    4639   4702#   4756    4836
```

```
                    4899#  4953   5033   5097#  5274#  5333   5418   5481#  5540   5625   5688#  5747   5832
                    5895#  5954   6033   6102#  6161   6246   6309#  6368   6453   6516#  6575   6660   6723#
                    6782   6867   6930#  6989   7074   7137#  7196   7281   7343#  7517#  7760#  7897#  8137#
                    8205   8274#  8362   8455#  8668#  8759   8834#  8960#  9136   9207   9277#  9453   9524
                    9594#  9770   9841   9911#  10087  10158  10228#  10404  10475  10545#  10721  10792  10862#
                   11038  11109  11179#  11355  11426  11496#  11672  11743  11813#  11989  12060  12130#  12306
                   12377  12447#  12623  12694  12764#  12940  13011  13081#  13257  13328  13398#  13574  13645
                   13713#  13856#  14480#  14698#
EROR0    034440   14808#  15237  15275
EROR01   034442   14811#
EROR1  = 035352    2281#  2478#  2675#  2872#  3070#  3287#  3520#  3717#  3914#  4111#  4308#  4505#  4702#
                    4899#  5097#  5141   5211   5274#  5481#  5688#  5895#  6102#  6309#  6516#  6723#  6930#
                    7137#  7343#  7445   7479   7517#  7682   7722   7760#  7850   7897#  8080   8137#  8294#
                    8455#  8515   8599   8668#  8834#  8896   8960#  9277#  9594#  9911#  10228#  10545#  10862#
                   11179#  11496#  11813#  12130#  12447#  12764#  13081#  13398#  13713#  13746  13778  13816  13856#
                   14384  14452  14480#  14658  14698#
EROR11= 001000       2#  2281   2478   2675   2872   3070   3287   3520   3717   3914   4111   4308   4505
                    4702   4899   5097   5274   5481   5688   5895   6102   6309   6516   6723   6930   7137
                    7343   7517   7760   7897   8137   8294   8455   8668   8834   8960   9277   9594   9911
                   10228  10545  10862  11179  11496  11813  12130  12447  12764  13081  13398  13713  13856
                   14480  14698
ERR      003244     845    858#   862
ERRMSG   006730    1598*  1616   1619#
ERRPC    003322     864    875#
ERRVEC=  000004     164#  1051   1052*  1054*  1057*
ERR1     034732   15066  15075#
ERTAB0   007064    1613   1653#
ERTAB1   007070    1655#  15083*  15086*
EXIT   = 000205     191#
EXITER   007036    1639   1644#
FLOAT    003156     838#   844
FRCPFW=  000002       1#
FY       003202     846#   850    855
GNS    = ****** U  1552   1555   1556   1557   1558   1559   1560   1561   1562   1563   1564   1565   1566
                    1567   1568   1569
HALTS    006760    1583   1630#
HELDAT   034312   14511  14517  14672#  15171
HERISR   035422   15163  15170#
HILIM    006042    1397*  1405   1421#
HT     = 000011      74#  1133   1174
IBAD0  = 000004       1#
IBAD1  = 000005       1#
IBA16  = 000004       1#
IBA17  = 000010       1#
ICR    = 000012       1#
INCHAR   011424    1812   1840#
INIFLG   001506     391#   767    797    804*
INISR    034664    1903  15062#
INLP1    005756    1402#  1411
INNPR  = 000000       1#
INPUT  = 134415    1566#  1916   1967   1979   1987   2044   2052
INTTY    013520    1997   2016   2032   2201#
IOTVEC=  000020     169#
KMACTV   001470     380#   781    906*   907   1861   1875   1962*  2122*  2128*  2129*  2133   2158   2224*
                   2225
KMCM     011022     872   1769#
```

KMCR00	002100	536#											
KMCR01	002110	541#											
KMCR02	002120	546#											
KMCR03	002130	551#											
KMCR04	002140	556#											
KMCR05	002150	561#											
KMCR06	002160	566#											
KMCR07	002170	571#											
KMCR10	002200	576#											
KMCR11	002210	581#											
KMCR12	002220	586#											
KMCR13	002230	591#											
KMCR14	002240	596#											
KMCR15	002250	601#											
KMCR16	002260	606#											
KMCR17	002270	611#											
KMCSR	002066	519#	832*	847	853*	882	1012	1078	1691	1745	1879*	1888	1942
KMCSRH	002070	520#	1726*	1727*	1731*	1737*	1738*	1744*	1746*	1888*	1889*	1890	
KMCTL	002072	521#	1890*	1891*	1892								
KMNUM	001472	381#	735	989	1960*	1974*	2114*	2115	2123	2125			
KMPO4	002074	522#	1715*	1721	1758	1763	1892*	1893*	1894				
KMPO6	002076	523#	1732*	1894*	1895*								
KMRLVL	002060	516#	982	983*	1897*	1898*	1899	1904*					
KMRVEC	002056	515#	982*	1015	1880*	1881*	1897	1903*					
KMS100	002102	537#											
KMS101	002112	542#											
KMS102	002122	547#											
KMS103	002132	552#											
KMS104	002142	557#											
KMS105	002152	562#											
KMS106	002162	567#											
KMS107	002172	572#											
KMS110	002202	577#											
KMS111	002212	582#											
KMS112	002222	587#											
KMS113	002232	592#											
KMS114	002242	597#											
KMS115	002252	602#											
KMS116	002262	607#											
KMS117	002272	612#											
KMS200	002104	538#											
KMS201	002114	543#											
KMS202	002124	548#											
KMS203	002134	553#											
KMS204	002144	558#											
KMS205	002154	563#											
KMS206	002164	568#											
KMS207	002174	573#											
KMS210	002204	578#											
KMS211	002214	583#											
KMS212	002224	588#											
KMS213	002234	593#											
KMS214	002244	598#											
KMS215	002254	603#											
KMS216	002264	608#											
KMS217	002274	613#											

```
KHTLVL  002064  518#   984    985*   1901*  1902*  1906*
KHTVEC  002062  517#   984*   1899*  1900*  1901   1905*
KH.END  002300  616#   1958   2229
KH.ESP  002100  386    535#   739    806    816    1871   1873   1956   1961   2150   2227   2231
LDATI = 000010    18
LDATO = 000010    18
LDVRMT  035536  2275   2472   2669   2866   3064   3281   3514   3711   3908   4105   4302   4499   4696
                4893   5091   5288   5475   5682   5879   6096   6303   6510   6717   6924   7131   7337
                7511   7754   7891   8131   8288   8449   8662   8828   8954   9271   9588   9905   10222
                10539  10856  11173  11490  11807  12124  12441  12758  13075  13392  13707  13850  14474
                14692  15204#
LF    = 000012    75#   1168   1174
LINE    010545  1769#  2045
LOBITS  006046  1399*  1423#
LOCK    001444   367#  1089*  1430*  1607   15099*  15101
LOKFLG  001510   393#
LOLIM   006040  1396*  1407   1420#
LWLOOP= 000040    1#
MAINT = 000017    1#
MARHLD= 000000    1#
MARINC= 014000    1#   7946   7991   7994   8690   8727   8988   8991   8994   8997   9000   9003   9006
                9009   9012   9015   9018   9021   9024   9027   9030   9033   9036   9039   9042   9045
                9048   9051   9054   9057   9060   9063   9066   9069   9072   9075   9078   9081   9098
                9101   9160   9169   9172   9231   9305   9308   9311   9314   9317   9320   9323   9326
                9329   9350   9353   9356   9359   9362   9365   9368   9371   9374   9377   9380   9383
                9386   9389   9392   9395   9398   9401   9404   9407   9410   9413   9415   9418
                9477   9486   9489   9546   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   9732   9735   9738
                9803   9806   9865   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  10052  10111  10120
                10123  10162  10256  10259  10262  10265  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  10366  10369  10428  10437  10440
                10499  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  10683  10686  10745  10754  10757  10816
                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   11000   11003   11062   11071   11074   11133   11207
                11210   11213   11216   11219   11222   11225   11228   11231   11234   11237   11240   11243   11246
                11249   11252   11255   11258   11261   11264   11267   11270   11273   11276   11279   11282   11285
                11288   11291   11294   11297   11300   11317   11320   11379   11388   11391   11450   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   11634   11637   11696   11705   11708   11767   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   11951   11954   12013   12022   12025   12084   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   12271   12330   12339   12342   12401   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   12585   12588   12647   12656   12659   12718   12742   12745   12748   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
                12902   12905   12964   12973   12976   13035   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   13219
                13222   13281   13290   13293   13352   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   13536   13539
                13598   13607   13610   13669   13962   14355
MASKX   001454    374#
MASTEK  007767   1609    1769#
MCEND   036154  15220*  15221   15223*  15232   15234   15238*  15257*  15258   15260*  15269   15271   15274*  15306#
MCSREN  034664  15052#  15234   15238   15271   15274
MCSRX   007717    971    1769#
MCT1    014030   2276    2280#
MCT10   015730   3712    3716#
MCT11   016134   3909    3913#
MCT12   016340   4106    4110#
MCT13   016544   4303    4307#
MCT14   016750   4500    4504#
MCT15   017154   4697    4701#
MCT16   017360   4894    4898#
MCT17   017564   5092    5096#
MCT2    014234   2473    2477#
MCT20   017754   5269    5273#
MCT21   020170   5476    5480#
MCT22   020404   5683    5687#
MCT23   020620   5890    5894#
MCT24   021034   6097    6101#
MCT25   021250   6304    6308#
MCT26   021464   6511    6515#
MCT27   021700   6718    6722#
MCT3    014440   2670    2674#
MCT30   022114   6925    6929#
MCT31   022330   7132    7136#
MCT32   022544   7338    7342#
MCT33   022740   7512    7516#
MCT34   023210   7755    7759#
MCT35   023356   7892    7896#
MCT36   023642   8132    8136#
```

```
MCT37   024020          8289    8293#
MCT4    014644          2867    2871#
MCT40   024200          8450    8454#
MCT41   024416          8663    8667#
MCT42   024600          8829    8833#
MCT43   024734          8955    8959#
MCT44   025262          9272    9276#
MCT45   025610          9589    9593#
MCT46   026136          9906    9910#
MCT47   026464         10223   10227#
MCT5    015060          3065    3069#
MCT50   027012         10540   10544#
MCT51   027340         10857   10861#
MCT52   027666         11174   11178#
MCT53   030214         11491   11495#
MCT54   030542         11808   11812#
MCT55   031070         12125   12129#
MCT56   031416         12442   12446#
MCT57   031744         12759   12763#
MCT6    015270          3282    3286#
MCT60   032272         13076   13080#
MCT61   032620         13393   13397#
MCT62   033146         13708   13712#
MCT63   033314         13851   13855#
MCT64   034120         14475   14479#
MCT65   034350         14693   14697#
MCT7    015524          3515    3519#
MDATA   011154          1491    1501    1786#
MEMLIM  001466           379#    930*
MEPASS  007572           970    1769#
MERRPC  010044          1612    1769#
MERRX   007744           977    1769#
MERR2   007617          1769#   2135
MERR3   007644           903    1769#
MICPC = 000511          2281#   2331    2344    2411    2424    2447    2478#   2528    2541    2608    2621    2644    2675#
                        2725    2738    2805    2818    2841    2872#   2922    2935    3002    3015    3038    3070#   3124
                        3142    3218    3231    3254    3287#   3355    3368    3453    3466    3489    3520#   3570    3583
                        3650    3663    3686    3717#   3767    3780    3847    3860    3883    3914#   3964    3977    4044
                        4057    4080    4111#   4161    4174    4241    4254    4277    4308#   4358    4371    4438    4445
                        4474    4505#   4555    4568    4635    4648    4671    4702#   4752    4765    4832    4845    4906
                        4899#   4949    4962    5029    5042    5065    5097#   5137    5147    5207    5217    5243    5257
                        5329    5342    5414    5427    5450    5481#   5536    5549    5621    5634    5657    5680#   5749
                        5756    5820    5841    5854    5905#   5960    5963    6035    6048    6071    6102#   6157    6172
                        6242    6255    6278    6309#   6364    6377    6449    6462    6485    6516#   6571    6584    6656
                        6669    6682    6723#   6778    6791    6863    6876    6899    6930#   6985    6998    7070    7083
                        7106    7137#   7160    7205    7277    7290    7313    7343#   7441    7475    7488    7517#   7678
                        7718    7728    7750#   7846    7865    7897#   8076    8104    8137#   8201    8214    8264    8294#
                        8358    8371    8454    8465#   8511    8521    8595    8605    8637    8668#   8755    8768    8803
                        8834#   8882    8892    8926    8960#   9132    9145    9203    9216    9243    9277#   9449    9462
                        9520    9533    9560    9594#   9766    9779    9837    9850    9877    9911#   10083   10096   10154
                       10167   10194   10228#  10400   10413   10471   10484   10511   10545#  10717   10730   10788   10801
                       10828   10862#  11034   11047   11105   11118   11145   11179#  11351   11364   11422   11435   11462
                       11496#  11668   11681   11739   11752   11779   11813#  11985   11998   12056   12069   12096   12130#
                       12302   12315   12373   12386   12413   12447#  12619   12632   12690   12703   12730   12764#  12936
                       12949   13007   13020   13047   13081#  13253   13266   13324   13337   13364   13398#  13570   13583
                       13641   13654   13681   13713#  13742   13774   13812   13822   13856#  14053   14063   14073   14083
```

```
                    14093  14103  14113  14125  14135  14145  14155  14165  14175  14185  14195  14211  14230
                    14243  14256  14269  14282  14295  14308  14321  14334  14380  14408  14448  14480#  14654
                    14664  14698#  14807#  15041
MILINT  011032       1769#  15072
MILK    001504        387#   740#   979  1869#  1874#  1878
MLDER   036156      15266  15307#
MLOCK   007670        944   1769#
MLRLD = 010000         1#   7940   7961   8036   8045   8092   8148   8254   8305   8414   8462   8531   8555
                     8615   8675   8715   8761   8774   8841   8961   9088   9138   9151   9157   9209   9222
                     9278   9405   9455   9468   9474   9526   9539   9595   9722   9772   9785   9791   9843
                     9856   9912  10039  10069  10102  10108  10160  10173  10229  10356  10406  10419  10425
                    10477  10490  10546  10673  10723  10736  10742  10794  10807  10863  10940  11040  11053
                    11059  11111  11124  11180  11307  11357  11370  11376  11428  11441  11447  11624  11674
                    11687  11693  11745  11758  11814  11941  11991  12004  12010  12062  12075  12131  12258
                    12308  12321  12327  12379  12392  12448  12575  12625  12638  12644  12698  12709  12765
                    12892  12942  12955  12961  13013  13026  13082  13209  13259  13272  13278  13330  13343
                    13399  13526  13576  13589  13595  13647  13660  13870  14343  14484
MNEM    007771       898   1769#
MNPDT   023572      7925   7931   8112#
MODU    010433      1769#  2015
MPASSX  007733       975   1769#
MPFAIL  007534      1687   1704   1769#
MPRLD = 004000         1#   7943   7964   8039   8048   8151   8257   8308   8417   8465   8534   8558   8618
                     8678   8718   8844   8964   9281   9598   9915  10232  10549  10866  11183  11500  11617
                    12134  12451  12768  13085  13402  13873  14346  14487
MR      007614       951   1769#  1932
MRESET= 004000       191#
MSTCLR= 104410      1561#  1693
MTITLE  001000       237#   771
MTSTN   007755      1610   1769#  1917
MVECX   007725       973   1769#
NEXT    001442       366#  1429   2273#  2279   2470#  2476   2667#  2673   2864#  2870   3062#  3068   3279#
                    3285   3512#  3518   3709#  3715   3906#  3912   4103#  4109   4300#  4306   4497#  4503
                    4694#  4700   4891#  4897   5089#  5095   5266#  5272   5473#  5479   5680#  5686   5887#
                    5893   6094#  6100   6301#  6307   6508#  6514   6715#  6721   6922#  6928   7129#  7135
                    7336#  7341   7509#  7515   7722#  7728   7889#  7895   8129#  8135   8286#  8292   8447#
                    8453   8660#  8666   8826#  8832   8952#  8958   9269#  9275   9586#  9592   9903#  9909
                   10220#  10226  10537#  10543  10654#  10860  11171#  11177  11488#  11494  11805#  11811  12122#
                   12128  12439#  12445  12756#  12762  13073#  13079  13390#  13396  13705#  13711  13848#  13854
                   14472#  14478  14690#  14696  15063  15075#  15111#  15131#  15141#  15146#  15181#  15220  15257
                   15300
NORCT   010703       733   1769#  1863
NODEV   003240       834    856#
NPDAT   022702      7377   7383   7493#
NPRBYT= 000200         1#
NPRC  = 000010         1#
NPRDTI  023152      7569   7575   7736#
NUM     010275      1769#  1968
NXMEM = 000001         1#
OBADD = 000006         1#
OBAD1 = 000007         1#
OBA16 = 000004         1#
OBA17 = 000010         1#
OCR   = 000011         1#
OK      003220       848    851#   874
ONE     001464       378#
```

```
OTMPR = 000020            1#
OUISR   005360         1905   15162#
PACT00  002302          622#
PACT01  002306          625#
PACT02  002312          628#
PACT03  002316          631#
PACT04  002322          634#
PACT05  002326          637#
PACT06  002332          640#
PACT07  002336          643#
PACT10  002342          646#
PACT11  002346          649#
PACT12  002352          652#
PACT13  002356          655#
PACT14  002362          658#
PACT15  002366          661#
PACT16  002372          664#
PACT17  002376          667#
PAGE0 = 000000            1#  14456   14876   14946   15001
PAGE1 = 001000            1#
PAGE2 = 002000            1#
PAGE3 = 003000            1#
PARBIT= 040000          191#
PCLK  = 000020            1#
PERFOR= 004537          191#
PFTAB   007276         1688    1710#
PIR0  = 177772           81#
PIR0VE= 000240          175#
POPR0 = 012600          186#  1643
POV1SP= 005726          184#
POP2SP= 022626          188#  15129
PRIO    010374         1769#  1996
PRIRTY  013772         2217*  2223*  2236   2257#
PR0   = 000000            98#  2193   2194
PR1   = 000040            99#
PR2   = 000100           100#
PR3   = 000140           101#
PR4   = 000200           102#  1907   2195   15093
PR5   = 000240           103#  1904   1906   2196
PR6   = 000300           104#  2197
PR7   = 000340           105#   210    212    214    216    2198
PS    = 177776            78#    79    732*   941*  1907*  2162*  2172*  15093*
PSW   = 177776            79#
PUSHR0= 010046          185#  1640
PUSH1S= 005746          183#
PUSH2S= 024646          187#
PWFAIL  015134         15070   15111#
PWRVEC= 000024          170#  1666*  1667*  1676*  1682*  1701*  1702*
QV.FLG  001511          394#   738*   988*
R0B0  = 000200            1#
RACT  = 000100            1#
RBCC  = 000001            1#
RBR   = 000200            1#
RDCHR = 104402         1282    1555#
RDLIN = 104403         1355    1556#
RDOCT = 104404         1404    1557#
```

```
RESOS  = 104407  1560#  1629
RETN1    035330  15074  15098  15106  15145#
RETN2    035316  15139  15140  15141  15143#
RET1     035506  15169  15174  15183#
RET2     035516  15170  15185#
ROMCLK= 104412  1563#  1716  1719  1756  1761
RONPR  = 000001  1#
RNDY   = 000020  1#
RUN      001500  384#  741*  1867*  1868*  1875
SAVACT   001474  382#  901  2133*  2225*
SAVNUM   001476  383#  735*  986*  989*  2126*  2219*
SAVPC    001460  376#  862*  877  1436*  1655  15086
SAVSP    001456  375#
SAVOS  = 104406  1559#  1588
SCOPI  = 104405  1558#  15100
SCPDT    035106  15068  15099#
SCPDT1   035126  15102  15104#
SCPE   = 035500  2281#  2451  2478#  2648  2675#  2845  2872#  3042  3070#  3258  3287#  3493  3520#
                 3690  3717#  3887  3914#  4084  4111#  4281  4308#  4478  4505#  4675  4702#  4872
                 4899#  5069  5097#  5247  5274#  5454  5481#  5651  5688#  5860  5895#  6075  6102#
                 6282  6309#  6489  6516#  6696  6723#  6903  6930#  7110  7137#  7317  7343#  7489
                 7517#  7732  7760#  7869  7897#  8108  8137#  8324  8294#  8428  8455#  8641  8668#
                 8807  8834#  8930  8960#  9247  9277#  9564  9594#  9881  9911#  10198  10228#  10515
                 10545#  10832  10862#  11149  11179#  11466  11496#  11783  11813#  12100  12130#  12417  12447#
                 12734  12764#  13051  13081#  13368  13398#  13685  13713#  13826  13856#  14412  14480#  14668
                 14698#
SCPE0    034570  14954#  15045
SCPE1  = 001130  2#  2281  2478  2675  2872  3070  3287  3520  3717  3914  4111  4308  4505
                 4702  4899  5097  5274  5481  5688  5895  6102  6309  6516  6723  6930  7137
                 7343  7517  7760  7897  8137  8294  8455  8668  8834  8960  9277  9594  9911
                 10228  10545  10862  11179  11496  11813  12130  12447  12764  13081  13398  13713  13856
                 14480  14698
SCP1   = 035426  2281#  2478  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  10100  10171  10228#  10417  10488  10545#  10734  10805  10862#
                 11051  11122  11179#  11368  11439  11496#  11585  11756  11813#  12002  12073  12130#  12319
                 12390  12447#  12636  12707  12764#  12953  13024  13081#  13270  13341  13398#  13587  13658
                 13713#  13856#  14480#  14698#
SCP10    034516  14884#
SCP11  = 035430  2281#  2478#  2675#  2872#  3070#  3287#  3520#  3717#  3914#  4111#  4308#  4505#  4702#
                 4899#  5097#  5151  5221  5274#  5481#  5688#  5895#  6102#  6309#  6516#  6723#  6930#
                 7137#  7343#  7517#  7760#  7897#  8137#  8294#  8455#  8525  8609  8668#  8834#  8896
                 8960#  9277#  9594#  9911#  10228#  10545#  10862#  11179#  11496#  11813#  12130#  12447#  12764#
                 13081#  13398#  13713#  13856#  14480#  14698#
SCP110   034520  14887#
SCP111= 001056  2#  2281  2478  2675  2872  3070  3287  3520  3717  3914  4111  4308  4505
                 4702  4899  5097  5274  5481  5688  5895  6102  6309  6516  6723  6930  7137
                 7343  7517  7760  7897  8137  8294  8455  8668  8834  8960  9277  9594  9911
```

```
                    10220  10545  10862  11179  11496  11813  12130  12447  12764  13081  13398  13713  13856
                    14480  14698
SKOPE   035452      15165  15175#
SOFTSW  011456       1810   1849#
SPACNT= 006401       1471#  1495   1498#  1512#
STACK = 001200         69#   733    920   2278   2475   2672   2869   3067   3284   3517   3714   3911   4108
                     4305   4502   4699   4896   5094   5271   5478   5685   5892   6099   6306   6513   6720
                     6927   7134   7340   7514   7757   7894   8134   8291   8452   8665   8831   8957   9274
                     9591   9808  10225  10542  10859  11176  11493  11810  12127  12444  12761  13078  13395
                    13710  13853  14477  14695  15092  15137  15301
STAT    001450       372#
STAT1   002050       508#  1882#
STAT2   002052       509#  1883#
STAT3   002054       510#  1884#
STKLMT= 177774        80#
STRTSW  001446       371#   773#  776#   777    779    793#   799    801    896    942    949   1910   1933#
                    1963   2146
SVLPAD  035534      15164#  15167  15175  15188#
SVLPDR  035354      15064#  15083  15147  15149#
SVNXT   035356      15063#  15091  15143  15146  15150#
SVOS    006070       1440#
SWFLG   011422        736#  1805#  1832#  1838#
SWR     001240        267#   757#   759    763#   773    901    906   1046   1064   1087   1577   1582   1638
                     1645   1647   1674   1694#  1734   1794   1831#  15087  15089
SWREG   000176        230#   752#   763    1794   1851
SW0   = 000001        133#
SW00  = 000001        123#   133    777    793   1963   2146
SW01  = 000002        122#   132    949   1910   1933
SW02  = 000004        121#   131
SW03  = 000010        120#   130    896
SW04  = 000020        119#   129
SW05  = 000040        118#   128
SW06  = 000100        117#   127   1734
SW07  = 000200        116#   126
SW08  = 000400        115#   125   1645  15087
SW09  = 001000        114#   124   1087
SW1   = 000002          1#   132#
SW10  = 002000        113#  1647  15089
SW11  = 004000        112#
SW12  = 010000        111#  1577
SW13  = 020000        110#  1582
SW14  = 040000        109#
SW15  = 100000        108#
SW2   = 000004          1#   131#
SW3   = 000010        130#
SW4   = 000020        129#
SW5   = 000040        128#
SW6   = 000100        127#
SW7   = 000200        126#
SW8   = 000400        125#
SW9   = 001000        124#
TBITVE= 000014        166#
TEMP    011112       1478   1742*  1747*  1753*  1765*  1784#  15112*  15117#
TIMER = 104414       1565#
TKVEC = 000060        173#
TLAST = 034314       1936  15312#
```

```
TPVEC = 000064            174#
TRAPVE= 000034            172#
TRTVEC= 000014            167#
TSTEN   034440          14789#  15223   15260
TSTRT   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    7888#
TST36   023606           7889    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   13947#
TST64   034064          13948   14471#
```

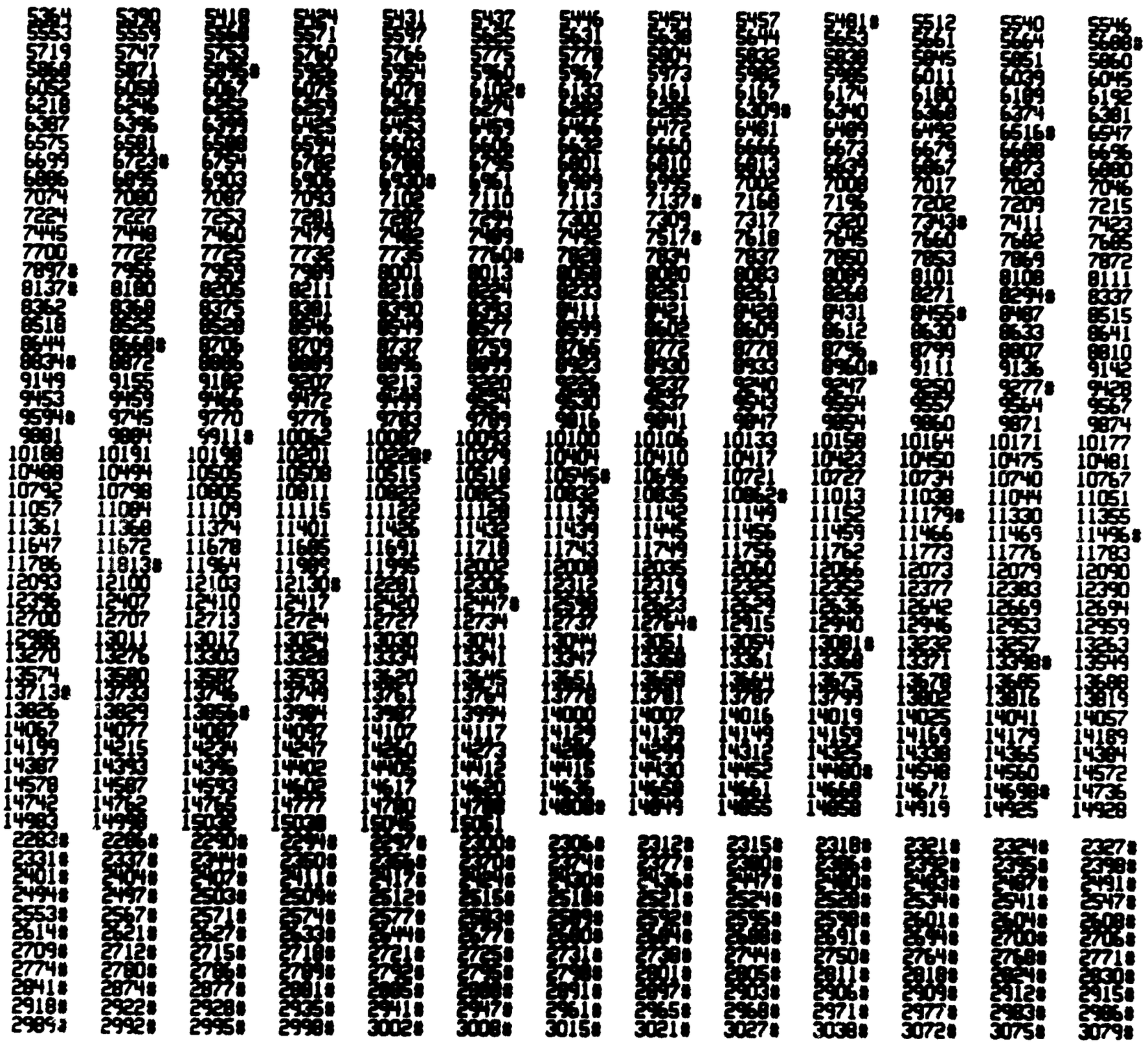
```
TST65   034314          14472   14689#  15312
TST66 = ###### U        14690
TST7    015470           3279    3511#
TTST    004146            945#    947#  1042#
TWOSYN= 010000            191#
TYPDAT  006744           1604    1622   1625#
TYPE  = 104401            771     783    805     860     865     871     898     903     944     951     970     971     973
                          975     977   1138    1289    1295    1300    1304    1309    1310    1312    1315    1319    1384
                         1386    1402   1409    1410    1461    1501    1552#   1605    1606    1609    1610    1612    1614
                         1618    1623   1687    1703    1808    1811    1863    1915    1932    1938    1975    1995    2009
                         2014    2023   2030    2037    2135   15072   15266
TYPMSG  006644           1602    1605#
UPCSGR  034630          15009#
VEC     010353           1769#   1988
VECMAP  013234           2134    2146#
VECTR   013766           2216#   2222#   2241    2247#   2255#
VEC4  = 000100              1#
VERFY   035734          15255#
WATCLK  036066          15290#
WCRAM   035536          15205#
WHAT    005760           1395#   1403#
WHERE   006044           1398#   1412    1422#
WHICH   013226           1977    2142#
WRDCNT  006376           1469#   1502#   1510#
WRKO.F  006732           1617    1620#
XA80  = 000200              1#
XACT  = 000100              1#
XBX     006532           1578    1580    1582#
XCSR    004104            972    1010#
XEOM  = 000002              1#
XERR    004126            978    1019#
XHEAD   010051            805    1769#
XPASS   004120            976    1016#
XRDY  = 000020              1#
XSFU  = 000040              1#
XSOM  = 000001              1#
XSTATQ  011064            814    1769#
XTSTN   007072           1611    1656#
XVEC    004112            974    1013#
ZERO    001462            377#
$APTHD  002034            490     496#
$ASTAT= ###### U         1208    1223
$ATYC   004712           1179    1181#
$ATY1   004666           1177#
$ATY3   004674           1123    1178#
$ATY4   004704           1180#   1633
$AUTOB  001234            264#
$BASE   001372            339#   2221
$BDADR  001222            259#  15079#  15312
$BDDAT  001226            261#  15078#  15312
$COM1   001376            341#   2218
$COM2   001400            342#
$CHARC  004662           1140#   1150#   1157    1166#   1171#
$CKSUM= ###### U         1555
$CMTAG  001200            247#
$CM1  = 000006            279#    280#    281#    282#    283#    284#    285#
```

```
$CR2  = 000014  279#  280#  281#  282#  283#  284#  285#
$CR3  = 000006  277#  279
$CR4  = 000005  285#  286#  287#  288#  289#  290#
$CNTLG  005624  1330#
$CNTLU  005517  1304  1329#
$CPUOP  001344  313#
$CRAP = 177777  1#  2260#  2263  2267#  2457#  2460  2464#  2654#  2657  2661#  2851#  2854  2858#
                3048#  3051  3056#  3264#  3267  3273#  3499#  3502  3506#  3696#  3699  3703#  3893#
                3896  3900#  4090#  4093  4097#  4287#  4290  4294#  4484#  4487  4491#  4681#  4684
                4688#  4878#  4881  4885#  5075#  5078  5083#  5253#  5256  5260#  5460#  5463  5467#
                5667#  5670  5674#  5874#  5877  5881#  6081#  6084  6088#  6288#  6291  6295#  6495#
                6498  6502#  6702#  6705  6709#  6909#  6912  6916#  7116#  7119  7123#  7323#  7326
                7329#  7496#  7499  7503#  7739#  7742  7746#  7875#  7878  7883#  8117#  8120  8123#
                8274#  8277  8280#  8434#  8437  8441#  8647#  8650  8654#  8813#  8816  8820#  8936#
                8939  8946#  8953#  8956  8963#  9570#  9573  9580#  9807#  9890  9897#  10204#  10207
                10214#  10521#  10524  10531#  10838#  10841  10848#  11155#  11158  11165#  11472#  11475  11482#
                11789#  11792  11799#  12106#  12109  12116#  12423#  12426  12433#  12740#  12743  12750#  13057#
                13060  13067#  13374#  13377  13384#  13691#  13694  13699#  13832#  13835  13842#  14459#  14462
                14466#  14675#  14678  14683#
$CRLF  001313  292#  1139  1174  1309  1329  1389  1410  1461  1605  1606  1614  1915  1975
$DDMD  001402  343#  2226
$DDM1  001404  344#
$DDM10  001426  353#
$DDM11  001430  354#
$DDM12  001432  355#
$DDM13  001434  356#
$DDM14  001436  357#
$DDM15  001440  358#
$DDM2  001406  345#
$DDM3  001410  346#
$DDM4  001412  347#
$DDM5  001414  348#
$DDM6  001416  349#
$DDM7  001420  350#
$DDM8  001422  351#
$DDM9  001424  352#
$DEVCT  001326  304#
$DEVM  001374  340#  2224
$DDAGN  004100  987  996  1001  1007#
$ENDAD  004070  223  769  1003#  1636
$HDCT  004054  998#
$ERV  001336  309#  750  786  788  1118  1186  1210  1630  1796
$ERVM  001337  310#  791  1120  1125  1188
$ERP  003662  966#  14690  15221  15250
$ERPCT  004046  995#  999
$ERFLG  001203  250#  737*  969*  1031  1060  1062*  1082  1587*  1601  1615*  1689*
$ERMAX  001215  256#  1082
$ERROR  006502  213  1576#
$ERRPC  001216  257#  746*  990*  1039*  1584  1586*  1690*
$ERRTB  001512  411#  1597
$ERTTL  001212  254#  981  1021  1644*  1886*
$ETABL  001336  308#
$ETEND  001442  361#  502
$FATAL  001320  301#  1214*
$FFLG  005132  1177*  1180*  1208  1217*  1225#
$FILLC  001256  275#  1143  1174
```

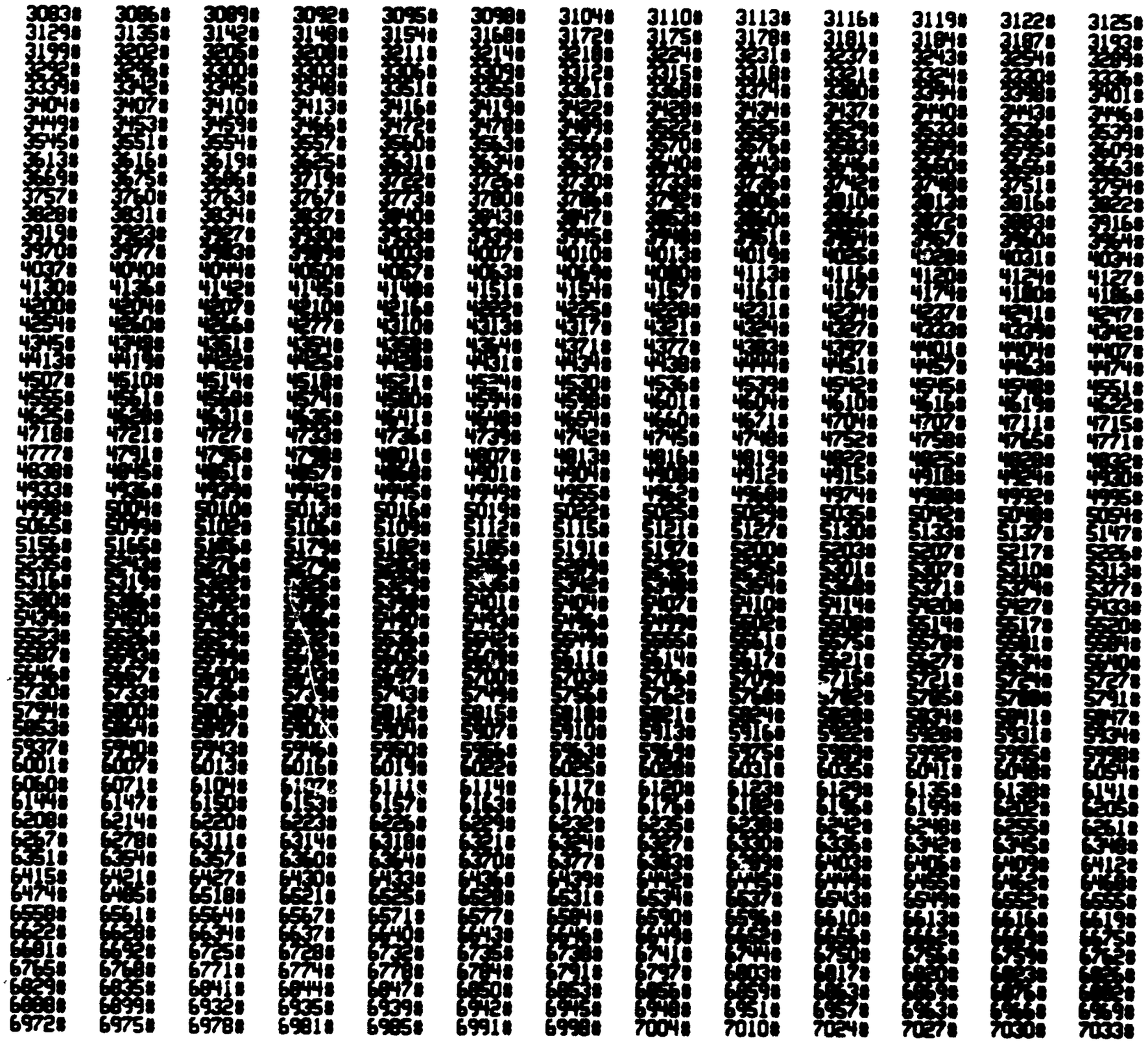
```
SFILLS  001255            274#  1174
SGDBDR  001220            258#  15082*  15312
SGDDAT  001224            260#  15077*  15312
SGETW2  004060           1000#
SGTSWP= ******  U        1554
SHD     = 000000           42
SHIBTS  002034            497#
SHIOCT  005712           1377*  1388#
SICNT   001204            251#  1068*  1069   1071*  1081
SILLUP  002270           1662   1682   1706#
SINPUT  005714           1393#  1566
SINTAG  001236            265#
SITEMB  001214            255#  1592*  1632
SLF     001314            293#  1174   1319   1329   1389
SLFLG   005131           1218*  1224#
SLPADR  001206            252#   948*   948#   952   1075*  1077   1081   1931*  1941*  1943   15064  15076*  15091*
                         15094  15113*  15130*  15140*  15143*  15144  15147*  15164  15165*  15167*  15175*  15181  15300*
                         15302
SLPERR  001210            253#
SMADR1  001350            326#
SMADR2  001354            330#
SMADR3  001360            333#
SMADR4  001364            336#
SMAIL   001316            299#   498    502    1074   1118
SMAMS1  001346            320#  2220
SMAMS2  001352            328#
SMAMS3  001356            331#
SMAMS4  001362            334#
SMBADR  002036            498#
SMFLG   005130           1178*  1184   1219*  1223#
SMMEM   005542           1333#  1811
SMSGAD  001332            306#  1194*  1197
SMSGLG  001334            307#  1199*
SMSGTY  001316            300#  1192   1200*  1212   1216*
SMSWR   005531           1331#  1808
SMTYP1  001347            321#
SMTYP2  001353            329#
SMTYP3  001357            332#
SMTYP4  001363            335#
SMXCNT  004356           1072   1081#
SN      = 000065            1#  2260   2267   2270   2275   2276   2280   2281#  2457   2464   2467   2472   2473
                         2477   2478#  2654   2661   2664   2669   2670   2674   2675#  2851   2858   2861   2866
                         2867   2871   2872#  3040   3056   3059   3064   3065   3069   3070#  3264   3273   3276
                         3281   3282   3286   3287#  3499   3506   3509   3514   3515   3519   3520#  3696   3703
                         3706   3711   3712   3716   3717#  3893   3900   3903   3908   3909   3913   3914#  4090
                         4097   4100   4105   4106   4110   4111#  4287   4294   4297   4302   4303   4307   4308#
                         4484   4491   4494   4499   4500   4504   4505#  4681   4688   4691   4696   4697   4701
                         4702#  4878   4885   4888   4893   4894   4898   4899#  5075   5083   5086   5091   5092
                         5096   5097#  5253   5260   5263   5268   5269   5273   5274#  5460   5467   5470   5475
                         5476   5480   5481#  5667   5674   5677   5682   5683   5687   5688#  5874   5881   5884
                         5889   5890   5894   5895#  6081   6088   6091   6096   6097   6101   6102#  6288   6295
                         6298   6303   6304   6308   6309#  6495   6502   6505   6510   6511   6515   6516#  6702
                         6709   6712   6717   6718   6722   6723#  6909   6916   6919   6924   6925   6929   6930#
                         7116   7123   7126   7131   7132   7136   7137#  7323   7329   7332   7337   7338   7342
                         7343#  7496   7503   7506   7511   7512   7516   7517#  7739   7746   7749   7754   7755
                         7759   7760#  7875   7883   7886   7891   7892   7896   7897#  8117   8123   8126   8131
```

```
                    8132  8136  8137#  8274  8280  8283  8288  8289  8293  8294#  8434  8441  8444
                    8445  8450  8454  8455#  8647  8654  8657  8662  8663  8667  8668#  8813  8820
                    8823  8828  8829  8833  8834#  8936  8946  8949  8954  8955  8959  8960#  9253
                    9263  9266  9271  9272  9276  9277#  9570  9580  9583  9588  9589  9593  9594#
                    9897  9899  9900  9905  9906  9910  9911#  10204  10214  10217  10222  10223  10227
                   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  12135  12139  12130#  12423  12433  12436  12441  12442  12446  12447#  12740  12750
                   12753  12758  12759  12763  12764#  13057  13067  13070  13075  13076  13080  13081#  13374
                   13384  13387  13392  13393  13397  13398#  13691  13699  13702  13707  13708  13712  13713#
                   13832  13842  13845  13850  13851  13855  13856#  14149  14166  14169  14174  14175  14179
                   14480#  14675  14683  14687  14692  14693  14697  14698#  15312#
SNULL   001254      273#  1145  1174
SOVER   004324     1044  1047  1058  1070  1076#
SPASS   001324      303#  968#  980  992#  993#  1010  1018  1066  1082  1885#
SPASTN  002042      500#
SPWRDN  007100      211  734  1666#  1701  15133  15136
SPWRUP  007264     1704#
SPWRUP  007152     1676  1682#
SQUES   001312      291#  1174  1312  1329  1386  1389  1409  1938  2010  2024  2038
SRDCHR  005134     1249#  1555
SRDDEC= ****** U   1558
SRDLIN  005254     1277#  1556
SRDOCT  005554     1350#  1557
SRDSZ = 000007     1270#
SREGA0  001260      277#
SREG0   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
$S      = 000067       1#  2273  2281#  2470  2478#  2667  2675#  2864  2872#  3062  3070#  3279  3287#
                    3512  3520#  3709  3717#  3906  3914#  4103  4111#  4300  4308#  4497  4505#  4694
                    4702#  4891  4899#  5089  5097#  5286  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#  7989  7997#  8129  8137#  8286  8294#  8447  8455#
                    8660  8668#  8826  8834#  8952  8960#  9269  9277#  9586  9594#  9903  9911#  10220
                   10228#  10537  10545#  10854  10862#  11171  11179#  11488  11496#  11805  11813#  12122  12130#
                   12439  12447#  12756  12764#  13073  13081#  13390  13398#  13705  13713#  13848  13856#  14472
                   14480#  14690  14698#
$SAVRE= ****** U   1558
$SAVR6  007274     1675#  1683  1684#  1685#  1708#
$SCOPE  004134      209  1038#  2190
$SETUP= 000000      991  1039  1238  1335
$SVLAD  004316     1055  1073#
$SVPC = 000040      221#  226
$SWR  = 164000        1#  42  290  291  963  991  1002  1008  1010  1032  1033  1034  1035
                    1046  1058  1060  1061  1062  1063  1064  1076  1081  1705
$SWREG  001340      311#  752
$SWRNK= 000000     1035
$TESTN  001322      302#  1074#
$TIMES  001210      290#  991#  1063#  1069  1072#  1081
```

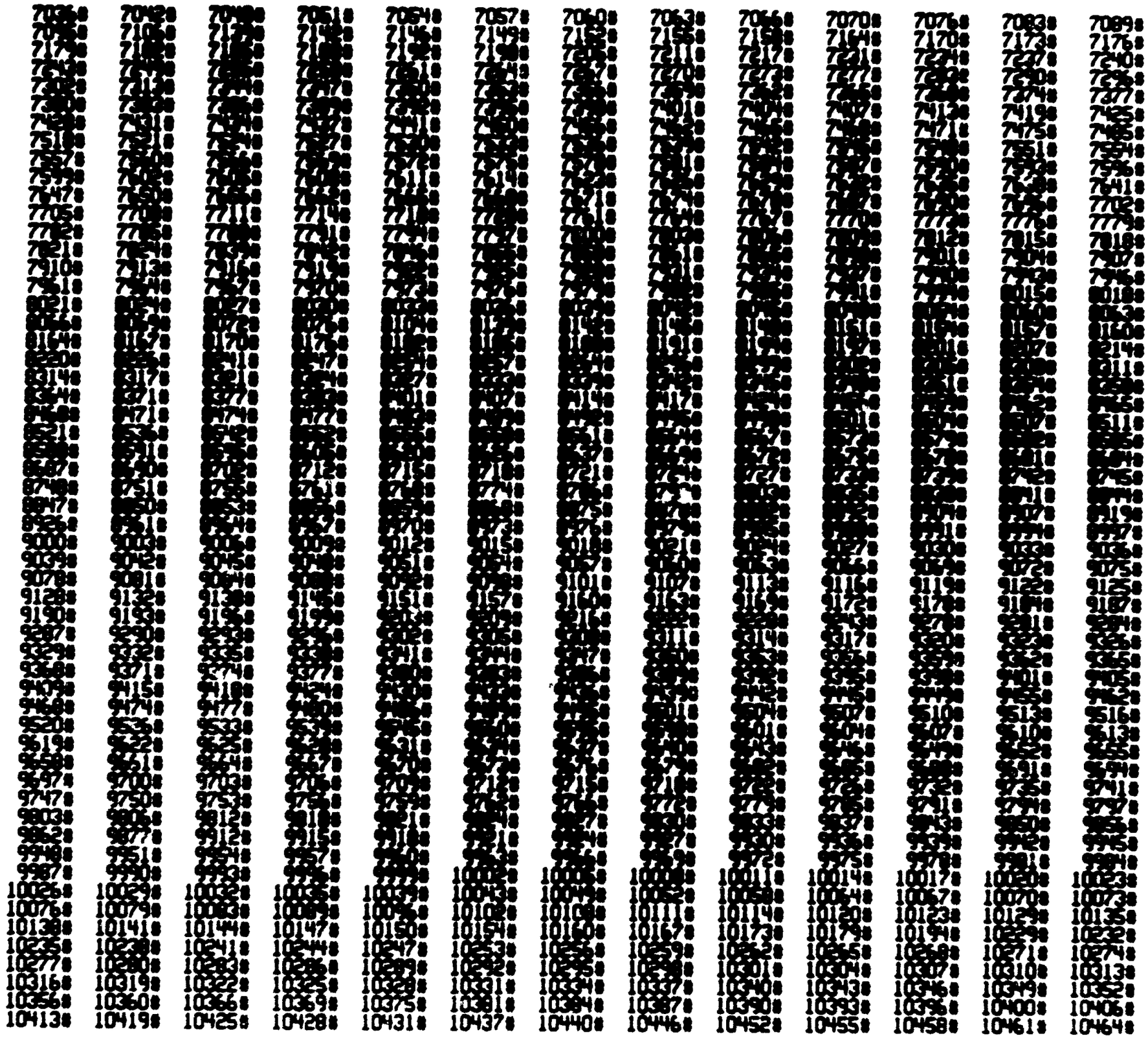
```
STKB    001246          270#  1045  1236  1253  1259  1798  1800  1842  2203
STKS    001244          269#  1043  1236  1251  1257  1840  2201
STMP0   001276          285#   807*  1771  2158*  2159*
STMP1   001300          286#   808*  1773
STMP2   001302          287#   810*   831*   858   867*  1775  1971  1974  2061*
STMP3   001304          288#   811*  1777  1983  1986  1991  1994  2048  2051  2056  2059
STMP4   001306          289#   812*  1779  1966*  1978*  2144
STN   = 000001            1#    42
STPB    001252          272#  1163*  1174  1581*  1845*  2206*
STPFLG  001257          276#  1112  1174
STPS    001250          271#  1161  1174  1579  1843  2204
STRAP   006404          215  1529#
STRAP2  006426         1540#  1551
STRP  = 000021         1544#  1553#  1555  1556#  1557#  1558#  1559#  1560#  1561#  1562#  1563#  1564#  1565#
                       1566#  1567#  1568#  1569#  1570#
STRPAD  006440         1534  1551#
STSTM   002040          499#
STSTNM  001202          249#   747*  1031  1073*  1074  1076  1082  1658  1712  1920  1927  1929  2272*
                       2469*  2666*  2863*  3061*  3278*  3511*  3708*  3905*  4102*  4299*  4496*  4693*  4890*
                       5088*  5265*  5472*  5679*  5886*  6093*  6300*  6507*  6714*  6921*  7128*  7334*  7508*
                       7751*  7888*  8120*  8285*  8446*  8659*  8825*  8951*  9268*  9585*  9902*  10219*  10536*
                      10853*  11170*  11487*  11804*  12121*  12438*  12755*  13072*  13389*  13704*  13847*  14471*  14689*
STTYIN  005510         1279  1280  1292  1310  1324  1328#
STYPBN= ****** U       1553
STYPOS= ****** U       1553
STYPE   004404         1112#  1205  1544  1552
STYPEC  004616         1142  1149  1156  1161#  1162
STYPEX  004664         1167  1169  1172#
STYPOC= ****** U       1553
SUNIT   001330          305#
SUNITM  002044          501#
SUSWR   001342          312#
SVECT1  001366          337#  2222  2223
SVECT2  001370          338#
SXTSTR  004174         1041  1049#
SY    = 000000            1#   503#
$$GET4= 000000         1002#
$$INIT= 033440         2281#  2310  2335  2341  2348  2354  2363  2366  2390  2415  2421  2428  2434
                       2443  2451  2454  2478#  2507  2532  2538  2545  2551  2560  2563  2587  2612
                       2618  2625  2631  2640  2648  2661  2675#  2704  2729  2735  2743  2748  2757
                       2760  2784  2809  2815  2822  2828  2837  2845  2848  2872#  2901  2926  2932
                       2939  2945  2954  2957  2981  3006  3012  3019  3035  3034  3042  3045  3070#
                       3108  3133  3139  3146  3152  3161  3164  3197  3222  3228  3235  3241  3250
                       3258  3261  3287#  3334  3359  3365  3372  3378  3387  3390  3432  3457  3463
                       3470  3476  3485  3493  3496  3520#  3549  3574  3580  3587  3593  3602  3605
                       3629  3654  3660  3667  3673  3682  3690  3693  3717#  3746  3771  3777  3784
                       3790  3799  3802  3826  3851  3857  3864  3870  3879  3887  3890  3914#  3943
                       3968  3974  3981  3987  3996  3999  4023  4048  4054  4061  4067  4076  4084
                       4087  4111#  4140  4165  4171  4178  4184  4193  4196  4220  4245  4251  4258
                       4264  4273  4281  4284  4308#  4337  4362  4368  4375  4381  4390  4393  4417
                       4442  4448  4455  4461  4470  4478  4481  4505#  4534  4559  4565  4572  4578
                       4587  4590  4614  4639  4645  4652  4658  4667  4675  4678  4702#  4731  4756
                       4762  4769  4775  4784  4787  4811  4836  4842  4849  4855  4864  4872  4875
                       4899#  4928  4953  4959  4966  4972  4981  4984  5008  5033  5039  5046  5052
                       5061  5069  5072  5097#  5125  5141  5144  5151  5154  5169  5172  5135  5211
                       5214  5221  5224  5239  5247  5250  5274#  5305  5333  5339  5346  5352  5361
```



SSSOER= 000001



CROSS REFERENCE TABLE -- USER SYMBOLS

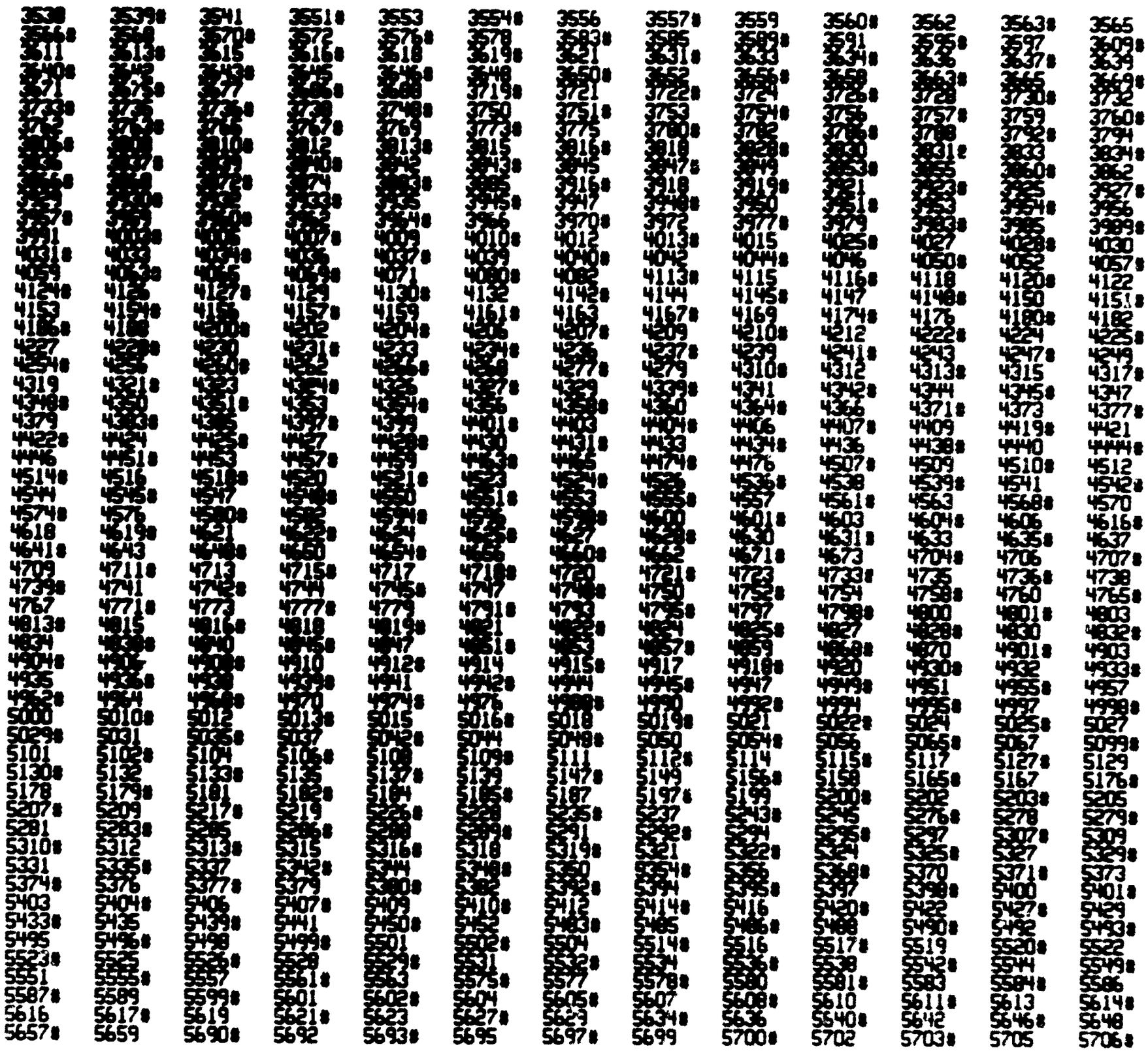


```
10467#  10471#  10477#  10480#  10490#  10496#  10511#  10516#  10519#  10522#  10525#  10550#  10561#
10564#  10570#  10573#  10576#  10577#  10580#  10583#  10586#  10591#  10594#  10597#  10600#  10603#
10606#  10609#  10612#  10615#  10616#  10617#  10620#  10623#  10626#  10627#  10630#  10633#  10642#
10645#  10646#  10651#  10654#  10655#  10656#  10657#  10660#  10663#  10664#  10667#  10670#  10673#
10676#  10677#  10680#  10683#  10684#  10685#  10686#  10687#  10690#  10691#  10694#  10695#  10698#
10701#  10704#  10707#  10710#  10711#  10712#  10713#  10714#  10715#  10716#  10717#  10720#  10721#
10724#  10727#  10730#  10733#  10734#  10735#  10736#  10737#  10740#  10741#  10742#  10743#  10744#
10747#  10750#  10753#  10756#  10757#  10760#  10763#  10766#  10767#  10770#  10771#  10774#  10775#
11027#  11030#  11033#  11036#  11037#  11040#  11043#  11046#  11047#  11050#  11053#  11056#  11057#
11060#  11063#  11066#  11067#  11070#  11073#  11076#  11077#  11100#  11103#  11106#  11107#  11110#
11113#  11116#  11117#  11120#  11123#  11126#  11127#  11130#  11133#  11136#  11137#  11140#  11143#
11146#  11147#  11150#  11153#  11156#  11157#  11160#  11163#  11166#  11167#  11170#  11173#  11176#
11177#  11200#  11203#  11206#  11207#  11210#  11213#  11216#  11217#  11220#  11223#  11226#  11227#
11230#  11233#  11236#  11237#  11240#  11243#  11246#  11247#  11250#  11253#  11256#  11257#  11260#
11263#  11266#  11267#  11270#  11273#  11276#  11277#  11300#  11303#  11306#  11307#  11310#  11313#
11316#  11317#  11320#  11323#  11326#  11327#  11330#  11333#  11336#  11337#  11340#  11343#  11346#
11347#  11350#  11353#  11356#  11357#  11360#  11363#  11366#  11367#  11370#  11373#  11376#  11377#
11402#  11405#  11406#  11411#  11414#  11415#  11420#  11423#  11426#  11427#  11430#  11433#  11436#
11437#  11442#  11445#  11446#  11451#  11454#  11455#  11460#  11463#  11466#  11467#  11472#  11475#
11476#  11501#  11504#  11507#  11510#  11511#  11514#  11517#  11520#  11521#  11524#  11527#  11530#
11531#  11534#  11537#  11540#  11541#  11544#  11547#  11550#  11551#  11554#  11557#  11560#  11561#
11564#  11567#  11570#  11573#  11574#  11577#  11600#  11603#  11606#  11607#  11610#  11613#  11616#
11617#  11622#  11625#  11626#  11631#  11634#  11635#  11640#  11643#  11646#  11647#  11652#  11655#
11656#  11661#  11664#  11665#  11670#  11673#  11674#  11677#  11702#  11705#  11706#  11711#  11714#
11715#  11720#  11723#  11724#  11727#  11730#  11733#  11734#  11737#  11740#  11743#  11744#  11747#
11752#  11755#  11756#  11761#  11764#  11765#  11770#  11773#  11774#  11777#  12002#  12005#  12010#
12013#  12014#  12017#  12022#  12025#  12026#  12031#  12034#  12035#  12040#  12043#  12046#  12047#
12052#  12055#  12056#  12061#  12064#  12065#  12070#  12073#  12074#  12077#  12102#  12105#  12106#
12111#  12114#  12115#  12120#  12123#  12124#  12127#  12132#  12135#  12136#  12141#  12144#  12145#
12150#  12153#  12154#  12157#  12162#  12165#  12166#  12171#  12174#  12175#  12200#  12203#  12206#
12207#  12212#  12215#  12216#  12221#  12224#  12225#  12230#  12233#  12234#  12237#  12242#  12245#
12246#  12251#  12254#  12255#  12260#  12263#  12264#  12267#  12272#  12275#  12276#  12301#  12304#
12307#  12312#  12315#  12316#  12321#  12324#  12325#  12330#  12333#  12334#  12337#  12342#  12345#
12346#  12351#  12354#  12355#  12360#  12363#  12364#  12367#  12372#  12375#  12376#  12401#  12404#
12407#  12412#  12415#  12416#  12421#  12424#  12425#  12430#  12433#  12434#  12437#  12442#  12445#
12446#  12451#  12454#  12455#  12460#  12463#  12464#  12467#  12472#  12475#  12476#  12501#  12504#
12507#  12512#  12515#  12516#  12521#  12524#  12525#  12530#  12533#  12534#  12537#  12542#  12545#
12546#  12551#  12554#  12555#  12560#  12563#  12564#  12567#  12572#  12575#  12576#  12601#  12604#
12607#  12612#  12615#  12616#  12621#  12624#  12625#  12630#  12633#  12634#  12637#  12642#  12645#
12646#  12651#  12654#  12655#  12660#  12663#  12664#  12667#  12672#  12675#  12676#  12701#  12704#
12707#  12712#  12715#  12716#  12721#  12724#  12725#  12730#  12733#  12734#  12737#  12742#  12745#
12746#  12751#  12754#  12755#  12760#  12763#  12764#  12767#  12772#  12775#  12776#  13001#  13004#
13007#  13012#  13015#  13016#  13021#  13024#  13025#  13030#  13033#  13034#  13037#  13042#  13045#
13046#  13051#  13054#  13055#  13060#  13063#  13064#  13067#  13072#  13075#  13076#  13101#  13104#
13107#  13112#  13115#  13116#  13121#  13124#  13125#  13130#  13133#  13134#  13137#  13142#  13145#
13146#  13151#  13154#  13155#  13160#  13163#  13164#  13167#  13172#  13175#  13176#  13201#  13204#
13207#  13212#  13215#  13216#  13221#  13224#  13225#  13230#  13233#  13234#  13237#  13242#  13245#
13246#  13251#  13254#  13255#  13260#  13263#  13264#  13267#  13272#  13275#  13276#  13301#  13304#
13307#  13312#  13315#  13316#  13321#  13324#  13325#  13330#  13333#  13334#  13337#  13342#  13345#
13346#  13351#  13354#  13357#  13360#  13363#  13366#  13370#  13376#  13383#
```

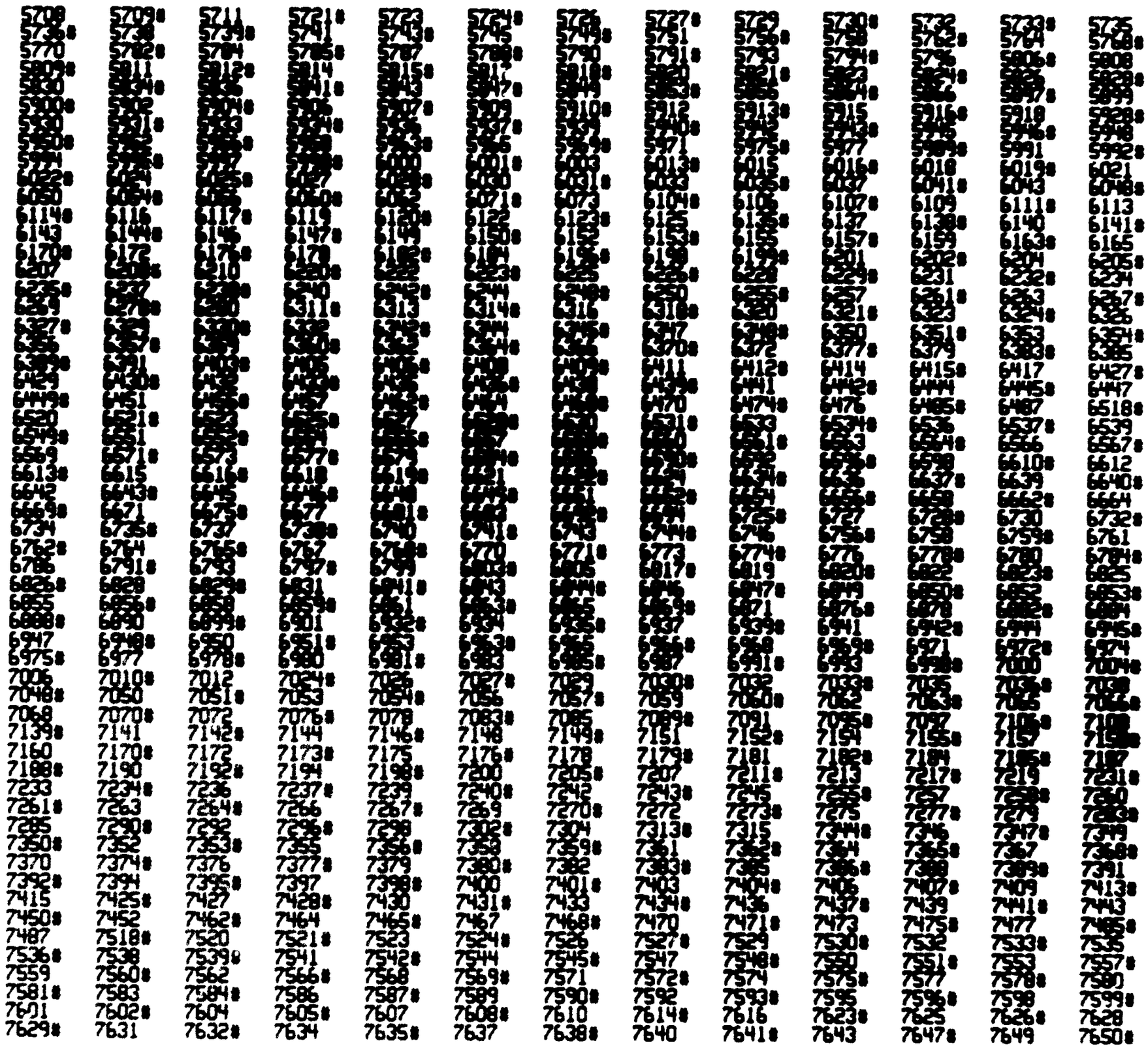
13589# 13595# 13598# 13601# 13607# 13610# 13616# 13622# 13625# 13628# 13631# 13634# 13637#
13641# 13647# 13650# 13660# 13666# 13681# 13714# 13717# 13720# 13723# 13726# 13729# 13735#
13738# 13742# 13751# 13754# 13757# 13767# 13770# 13774# 13783# 13795# 13800# 13803# 13812#
13822# 13827# 13840# 13843# 13846# 13870# 13873# 13877# 13878# 13882# 13886# 13890# 13892#
13895# 13900# 13910# 13913# 13919# 13912# 13915# 13918# 13921# 13927# 13930# 13933# 13936#
13939# 13942# 13948# 13951# 13961# 13964# 13967# 13972# 13971# 13974# 13980# 13990# 13996#
14009# 14021# 14027# 14031# 14034# 14046# 14049# 14053# 14059# 14063# 14069# 14073# 14079#
14083# 14089# 14093# 14098# 14103# 14109# 14113# 14121# 14125# 14131# 14136# 14141# 14145#
14151# 14156# 14161# 14165# 14171# 14175# 14181# 14186# 14191# 14196# 14204# 14207# 14211#
14217# 14220# 14223# 14226# 14230# 14236# 14239# 14243# 14249# 14252# 14256# 14262# 14265#
14269# 14275# 14278# 14282# 14288# 14291# 14295# 14301# 14304# 14308# 14314# 14317# 14321#
14327# 14330# 14334# 14343# 14346# 14349# 14352# 14356# 14361# 14367# 14370# 14373# 14376#
14380# 14400# 14417# 14420# 14423# 14426# 14429# 14438# 14441# 14444# 14448# 14481# 14484#
14487# 14490# 14493# 14496# 14499# 14502# 14505# 14508# 14511# 14514# 14517# 14520# 14523#
14526# 14529# 14532# 14535# 14538# 14541# 14544# 14550# 14553# 14556# 14562# 14565# 14568#
14574# 14580# 14583# 14589# 14595# 14598# 14604# 14607# 14613# 14623# 14626# 14632# 14638#
14641# 14644# 14647# 14650# 14654# 14664# 14699# 14702# 14705# 14708# 14711# 14714# 14717#
14720# 14723# 14726# 14729# 14732# 14738# 14747# 14750# 14753# 14758# 14761# 14804# 14863#
14821# 14824# 14827# 14830# 14833# 14836# 14839# 14842# 14845# 14851# 14860# 14815# 14818#
14866# 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#

SSSSER= 000001

2283# 2285 2286# 2288 2290# 2292 2294# 2296 2297# 2299 2300# 2302 2312#
2314 2315# 2317 2318# 2320 2321# 2323 2324# 2326 2327# 2329 2331# 2333
2337# 2339 2344# 2346 2350# 2352 2356# 2358 2370# 2372 2374# 2376 2377#
2379 2380# 2382 2392# 2394 2396# 2397 2398# 2400 2401# 2403 2404# 2406
2407# 2409 2411# 2413 2417# 2419 2424# 2426 2430# 2432 2436# 2438 2447#
2449 2480# 2482 2483# 2485 2487# 2489 2491# 2493 2494# 2496 2497# 2499
2509# 2511 2512# 2514 2515# 2517 2518# 2520 2521# 2523 2524# 2526 2528#
2530 2534# 2536 2541# 2543 2547# 2591 2614# 2616 2621# 2623 2627# 2569 2571# 2573
2574# 2576 2577# 2579 2589# 2610 2640# 2642 2644# 2646 2647# 2650 2633# 2601#
2603 2604# 2606 2608# 2670 2677# 2679 2680# 2682 2683# 2685 2691# 2693 2695#
2696 2706# 2708 2709# 2711 2712# 2714 2715# 2717 2718# 2720 2721# 2723
2725# 2727 2731# 2733 2738# 2740 2744# 2746 2750# 2762 2764# 2766 2768#
2770 2771# 2773 2774# 2776 2780# 2788 2791 2793 2795# 2797
2798# 2800 2801# 2803 2805# 2807 2811# 2813 2816# 2818# 2869 2880 2890
2832 2841# 2843 2874# 2876 2877# 2879 2881# 2883 2885# 2887 2915# 2917 2918#
2891# 2893 2903# 2905 2906# 2908 2910# 2911 2912# 2914 2947# 2949 2961# 2963
2920 2922# 2924 2926# 2930 2931# 2933 2935# 2937 2938# 2940 2941# 2943 2945#
2955# 2957 2960# 2970 2971# 2973 2975# 2977 2979# 2981 2983# 2985 2987# 2989
2994 2996# 2997 2998# 3000 3002# 3004 3006# 3008 3010 3012# 3014# 3017 3021#
3027# 3029 3038# 3040 3072# 3074 3075# 3077 3078# 3081 3083# 3085 3116#
3088 3089# 3091 3095# 3097 3098# 3100 3108# 3110# 3112# 3115 3142# 3144
3119# 3121 3122# 3124 3125# 3127 3128# 3131 3135# 3137 3178# 3180 3181#
3150 3154# 3156 3160# 3170 3172# 3174 3175# 3177 3208# 3210 3211# 3213
3187# 3189 3199# 3201 3202# 3204 3205# 3207 3227# 3243# 3245 3306# 3308
3216 3218# 3220 3224# 3226 3231# 3233 3300# 3302 3303# 3305 3341 3368#
3289# 3291 3292# 3294 3296# 3298# 3299# 3301# 3323 3355# 3357 3361# 3363 3342# 3370
3314 3315# 3317 3318# 3320 3321# 3353 3398# 3400 3401# 3403 3404# 3406 3410# 3412
3345# 3347 3348# 3350 3351# 3396 3419# 3421 3434# 3436 3437# 3439 3440# 3442 3443#
3376 3380# 3382 3394# 3418 3451 3453# 3455 3459# 3461 3466# 3468 3472# 3474
3413# 3415 3416# 3448 3449# 3478# 3480 3489# 3491 3522# 3524 3525# 3527 3529# 3531 3533# 3535 3536#
3445 3446#



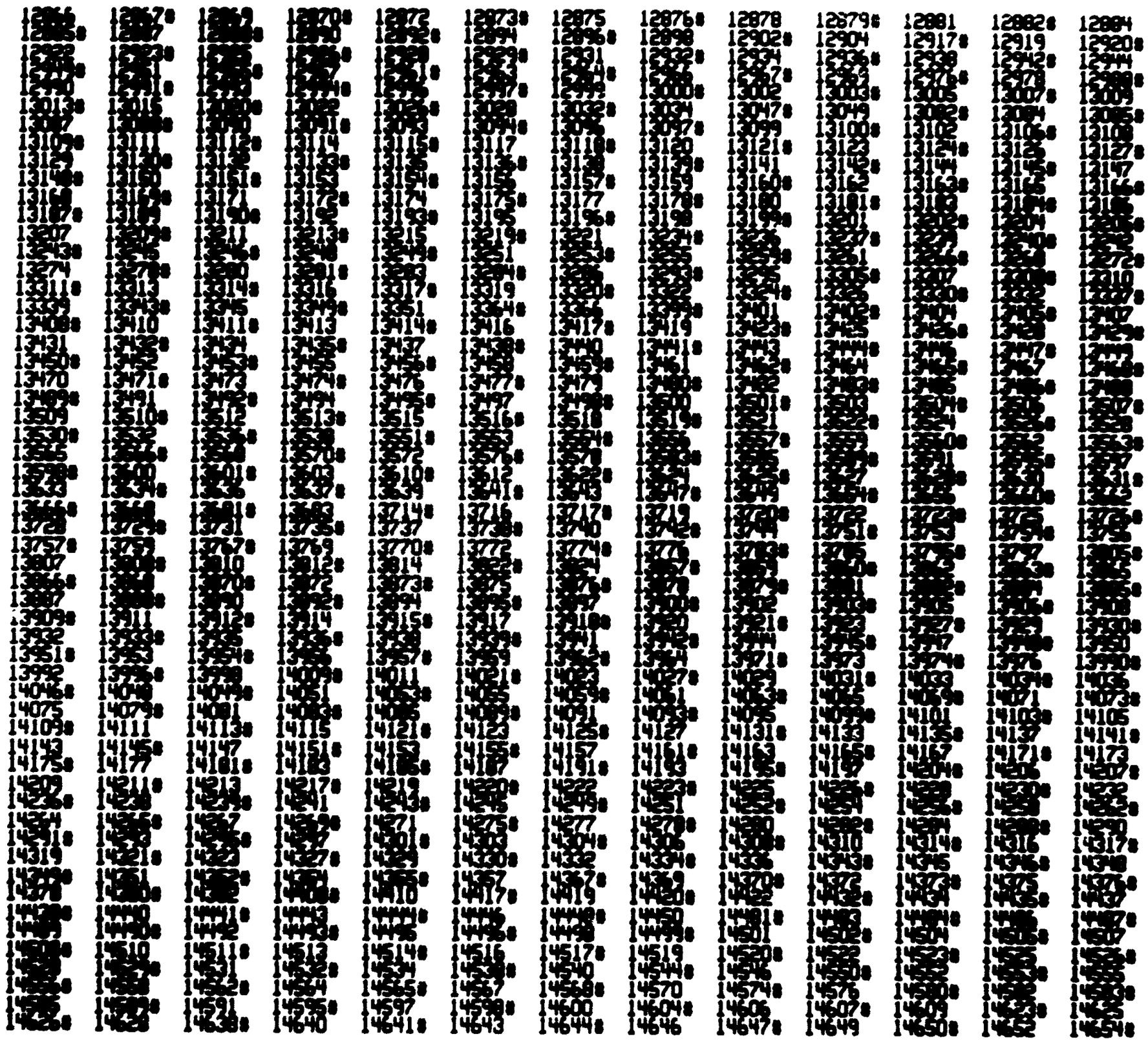
F10



```
7652    7662#   7664    7665#   7667    7668#   7670    7671#   7673    7674#   7676    7678#   7680
7687#   7689    7690#   7692    7702#   7704    7705#   7707    7708#   7710    7711#   7713    7714#
7716    7718#   7720    7728#   7730    7761#   7763    7764#   7766    7767#   7769    7770#   7772
7773#   7775    7776#   7778    7779#   7781    7782#   7784    7785#   7787    7788#   7790    7791#
7793    7794#   7796    7797#   7799    7800#   7802    7803#   7805    7806#   7808    7809#   7811
7812#   7814    7818#   7820    7821#   7823    7824#   7826    7839#   7841    7842#   7844    7846#
7848    7855#   7857    7858#   7860    7865#   7867    7898#   7900    7901#   7903    7904#   7906
7907#   7909    7910#   7912    7913#   7915    7916#   7918    7919#   7921    7922#   7924    7925#
7927    7928#   7930    7931#   7933    7934#   7936    7937#   7939    7940#   7942    7943#   7945
7946#   7948    7961#   7963    7964#   7966    7967#   7969    7970#   7972    7973#   7975    7976#
7978    7979#   7981    7982#   7984    7985#   7987    7991#   7993    7994#   7996    8015#   8017
8018#   8020    8021#   8023    8024#   8026    8027#   8029    8030#   8032    8033#   8035    8036#
8038    8039#   8041    8042#   8044    8045#   8047    8048#   8050    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    8182#   8184    8185#   8187    8188#   8190    8191#
8193    8194#   8196    8197#   8199    8201#   8203    8207#   8209    8214#   8216    8220#   8222
8226#   8228    8241#   8243    8254#   8256    8257#   8259    8264#   8266    8296#   8298    8299#
8301    8302#   8304    8305#   8307    8308    8310    8311#   8313    8314#   8316    8317#   8319
8321#   8323    8324#   8326    8327#   8328    8339#   8341    8342#   8344    8345#   8347    8348#
8350    8351#   8353    8354#   8356    8358    8360    8364#   8366    8371#   8373    8377#   8379
8383#   8385    8401#   8403    8414#   8416    8417#   8419    8424#   8426    8456#   8458    8459#
8461    8462#   8464    8465#   8467    8468    8470    8471#   8473    8474#   8476    8477#   8479
8489#   8491    8492#   8494    8495#   8497    8498#   8500    8501#   8503    8504#   8506    8507#
8509    8511#   8513    8521#   8523    8536#   8538    8552#   8554    8555#   8557    8558#   8560
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    8786#   8788    9803#   8805    8835#   8837    8838#   8840
8841#   8843    8844#   8846    8847#   8849    8850#   8852    8853#   8855    8856#   8858    8859#
8861    8868#   8870    8875#   8877    8878#   8880    8882#   8884    8892#   8894    8904#   8906
8907#   8909    8919#   8921    8936#   8938    8961#   8963    8964#   8966    8967#   8969    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    9240#   9242    9243#   9295    9296#   9298    9302#   9304    9305#   9307    9308#
9310    9311#   9313    9314#   9316    9317#   9319    9320#   9322    9323#   9325    9326#   9328
9329#   9331    9332#   9334    9335#   9337    9338#   9340    9341#   9343    9344#   9346    9347#
9349    9350#   9352    9353#   9355    9356#   9358    9359#   9361    9362#   9364    9365#   9367
9368#   9370    9371#   9373    9374#   9376    9377#   9379    9380#   9382    9383#   9385    9386#
9388    9389#   9391    9392#   9394    9395#   9397    9398#   9400    9401#   9403    9405#   9407
9409#   9411    9415#   9417    9430#   9432    9433#   9435    9436#   9438    9439#   9441    9442#
9444    9445#   9447    9449#   9451    9455#   9457    9462#   9464    9468#   9470    9474#   9476
9477#   9479    9480#   9482    9489#   9491    9501#   9503    9504#   9506    9507#   9509    9510#
9512    9513#   9515    9516#   9518    9520#   9522    9526#   9528    9533#   9535    9539#   9541
9545#   9547    9560#   9562    9595#   9597    9598#   9600    9601#   9603    9604#   9606    9607#
```

```
%09     %10#    %12     %13#    %15     %19#    %21     %22#    %24     %25#    %27     %28#    %30
%31#    %32#    %34#    %35#    %37#    %39#    %40#    %42#    %43#    %45#    %46#    %48#    %49#
%51     %52#    %54#    %55#    %57#    %58#    %60#    %61#    %63#    %64#    %66#    %67#    %69
%70#    %72#    %73#    %75#    %76#    %78#    %79#    %81#    %82#    %84#    %85#    %87#    %88#
%90#    %91#    %93#    %94#    %96#    %97#    %99#    %100#   %102#   %103#   %105#   %106#   %108
%109#   %111#   %112#   %114#   %115#   %117#   %118#   %120#   %121#   %123#   %124#   %126#   %127#
%129#   %130#   %132#   %133#   %135#   %136#   %138#   %139#   %141#   %142#   %144#   %145#   %147#
%148#   %150#   %151#   %153#   %154#   %156#   %157#   %159#   %160#   %162#   %163#   %165#   %166#
%168#   %169#   %171#   %172#   %174#   %175#   %177#   %178#   %180#   %181#   %183#   %184#   %186#
%187#   %189#   %190#   %192#   %193#   %195#   %196#   %198#   %199#   %201#   %202#   %204#   %205#
%207#   %208#   %210#   %211#   %213#   %214#   %216#   %217#   %219#   %220#   %222#   %223#   %225#
%226#   %228#   %229#   %231#   %232#   %234#   %235#   %237#   %238#   %240#   %241#   %243#   %244#
%246#   %247#   %249#   %250#   %252#   %253#   %255#   %256#   %258#   %259#   %261#   %262#   %264#
%265#   %267#   %268#   %270#   %271#   %273#   %274#   %276#   %277#   %279#   %280#   %282#   %283#
%285#   %286#   %288#   %289#   %291#   %292#   %294#   %295#   %297#   %298#   %300#   %301#   %303#
%304#   %306#   %307#   %309#   %310#   %312#   %313#   %315#   %316#   %318#   %319#   %321#   %322#
%324#   %325#   %327#   %328#   %330#   %331#   %333#   %334#   %336#   %337#   %339#   %340#   %342
%343#   %345#   %346#   %348#   %349#   %351#   %352#   %354#   %355#   %357#   %358#   %360#   %361#
%363#   %364#   %366#   %367#   %369#   %370#   %372#   %373#   %375#   %376#   %378#   %379#   %381#
%382#   %384#   %385#   %387#   %388#   %390#   %391#   %393#   %394#   %396#   %397#   %399#   %400#
%402#   %403#   %405#   %406#   %408#   %409#   %411#   %412#   %414#   %415#   %417#   %418#   %420#
%421#   %423#   %424#   %426#   %427#   %429#   %430#   %432#   %433#   %435#   %436#   %438#   %439#
%441#   %442#   %444#   %445#   %447#   %448#   %450#   %451#   %453#   %454#   %456#   %457#   %459#
%460#   %462#   %463#   %465#   %466#   %468#   %469#   %471#   %472#   %474#   %475#   %477#   %478#
%480#   %481#   %483#   %484#   %486#   %487#   %489#   %490#   %492#   %493#   %495#   %496#   %498#
%499#   %501#   %502#   %504#   %505#   %507#   %508#   %510#   %511#   %513#   %514#   %516#   %517#
%519#   %520#   %522#   %523#   %525#   %526#   %528#   %529#   %531#   %532#   %534#   %535#   %537#
%538#   %540#   %541#   %543#   %544#   %546#   %547#   %549#   %550#   %552#   %553#   %555#   %556#
%558#   %559#   %561#   %562#   %564#   %565#   %567#   %568#   %570#   %571#   %573#   %574#   %576#
%577#   %579#   %580#   %582#   %583#   %585#   %586#   %588#   %589#   %591#   %592#   %594#   %595#
%597#   %598#   %600#   %601#   %603#   %604#   %606#   %607#   %609#   %610#   %612#   %613#   %615#
%616#   %618#   %619#   %621#   %622#   %624#   %625#   %627#   %628#   %630#   %631#   %633#   %634#
%636#   %637#   %639#   %640#   %642#   %643#   %645#   %646#   %648#   %649#   %651#   %652#   %654#
%655#   %657#   %658#   %660#   %661#   %663#   %664#   %666#   %667#   %669#   %670#   %672#   %673#
%675#   %676#   %678#   %679#   %681#   %682#   %684#   %685#   %687#   %688#   %690#   %691#   %693#
%694#   %696#   %697#   %699#   %700#   %701#   %703#   %704#   %706#   %707#   %709#   %710#   %712#
%713#   %715#   %716#   %718#   %719#   %721#   %722#   %724#   %725#   %727#   %728#   %730#   %731#
%733#   %734#   %736#   %737#   %739#   %740#   %742#   %743#   %745#   %746#   %748#   %749#   %751#
%752#   %754#   %755#   %757#   %758#   %760#   %761#   %763#   %764#   %766#   %767#   %769#   %770#
%772#   %773#   %775#   %776#   %778#   %779#   %781#   %782#   %784#   %785#   %787#   %788#   %790#
%791#   %793#   %794#   %796#   %797#   %799#   %800#   %802#   %803#   %805#   %806#   %808#   %809#
%811#   %812#   %814#   %815#   %817#   %818#   %820#   %821#   %823#   %824#   %826#   %827#   %829#
%830#   %832#   %833#   %835#   %836#   %838#   %839#   %841#   %842#   %844#   %845#   %847#   %848#
%850#   %851#   %853#   %854#   %856#   %857#   %859#   %860#   %862#   %863#   %865#   %866#   %868#
%869#   %871#   %872#   %874#   %875#   %877#   %878#   %880#   %881#   %883#   %884#   %886#   %887#
%889#   %890#   %892#   %893#   %895#   %896#   %898#   %899#   %901#   %902#   %904#   %905#   %907#
%908#   %910#   %911#   %913#   %914#   %916#   %917#   %919#   %920#   %922#   %923#   %925#   %926#
%928#   %929#   %931#   %932#   %934#   %935#   %937#   %938#   %940#   %941#   %943#   %944#   %946#
%947#   %949#   %950#   %952#   %953#   %955#   %956#   %958#   %959#   %961#   %962#   %964#   %965#
%967#   %968#   %970#   %971#   %973#   %974#   %976#   %977#   %979#   %980#   %982#   %983#   %985#
%986#   %988#   %990#   %992#   %994#   %996#   11000#  11002   11015#  11017   11018#  11020   11021#
11023   11024#  11026   11027#  11029   11030#  11032   11034#  11036   11040#  11042   11047#  11049
11053#  11055   11059#  11061   11062#  11064   11065#  11067   11074#  11076   11086#  11088   11089#
11091   11092#  11094   11095#  11097   11098#  11100   11101#  11103   11105#  11107   11111#  11113
11118#  11120   11124#  11126   11130#  11132   11145#  11147   11180#  11182   11183#  11185   11186#
11188   11189#  11191   11192#  11194   11195#  11197   11198#  11200   11204#  11206   11207#  11209
11210#  11212   11213#  11215   11216#  11218   11219#  11221   11222#  11224   11225#  11227   11228#
11230   11231#  11233   11234#  11236   11237#
```

11230 11240# 11242 11243# 11245 11246# 11248 11249# 11251 11252# 11254 11256# 11257
11258# 11260 11261# 11263 11264# 11266 11267# 11269 11270# 11272 11273# 11275 11276#
11270 11272# 11301 11302# 11304# 11305# 11307# 11308 11310# 11312 11313# 11317# 11319 11320#
11327# 11329 11330 11332 11333 11335# 11337 11338# 11340 11341# 11343 11344# 11346 11347# 11349 11351# 11353
11357# 11359 11361# 11364# 11366 11370# 11372 11376# 11378 11379# 11381 11382# 11384 11391#
11340 11403# 11405# 11406# 11408 11436# 11437 11441# 11443 11447# 11449 11462# 11464 11467#
11422# 11424 11426# 11430 11432# 11433 11435 11436 11437 11441# 11443 11447# 11449 11462# 11464 11467#
11499 11500# 11502# 11504# 11505# 11506# 11507# 11508 11509# 11511 11512# 11514 11515# 11517
11521# 11523 11524# 11526 11527# 11529 11530# 11532 11533# 11536 11536# 11538 11539#
11541 11542# 11544 11545# 11547 11548# 11550 11551# 11553 11554# 11556 11557# 11559
11560# 11563 11564# 11566 11567# 11570 11571 11572# 11574 11575# 11577 11578#
11580 11581# 11583 11584# 11586 11587 11588# 11590# 11592 11593# 11595 11596# 11598
11599# 11601 11602# 11604 11605# 11607 11608# 11610 11611# 11613 11614# 11616 11617#
11619 11620# 11622 11624# 11626 11627 11630# 11634# 11636 11640# 11641 11642# 11644
11655# 11657 11660# 11662# 11663# 11665# 11666 11668# 11670 11674# 11676 11681#
11683 11687# 11690 11692# 11693# 11695# 11696 11698# 11701 11708# 11710 11720#
11723# 11725 11726# 11730# 11731 11732# 11734 11735# 11737 11739# 11741 11745#
11747 11750# 11751 11753# 11760 11764# 11766 11779# 11781 11814# 11816 11817# 11819
11820# 11822 11824# 11825# 11827 11828# 11830 11831 11832# 11834 11838# 11840 11841#
11843 11844# 11846# 11847# 11849# 11850# 11852 11853# 11855 11856# 11858 11859# 11861
11862# 11864 11866# 11867 11868# 11870 11871# 11873 11874# 11876 11877# 11879 11880#
11882 11883# 11885 11886# 11887# 11888# 11890# 11891 11892# 11894 11895# 11897 11898# 11900
11901# 11903 11904# 11906 11907# 11908# 11910# 11912 11913# 11915 11916# 11918 11919#
11921 11922# 11924 11925# 11927 11928# 11930 11931# 11933 11934# 11936 11937# 11939
11941# 11943 11945# 11947 11951# 11953 11956# 11958 11959# 11961 11962# 11964 11965#
11977 11978# 11980 11981# 11983 11985# 11987 11991# 11993 11998# 12000 12004# 12006
12010# 12012 12013# 12015 12016# 12018 12025# 12027 12037# 12039 12040# 12042 12043#
12045 12046# 12048 12049# 12051 12052# 12054 12056# 12058 12062# 12064 12069# 12071
12075# 12077 12081# 12083 12096# 12098 12131# 12133 12134# 12136 12137# 12139 12140#
12142 12143# 12145 12146# 12148 12149# 12151 12155# 12157 12158# 12160 12161# 12163
12164# 12166 12167# 12169# 12170# 12172 12173# 12175 12176# 12178 12179# 12181 12182#
12184 12185# 12187 12188# 12190 12191# 12193 12194# 12196 12197# 12199 12200# 12202
12203# 12205 12206# 12208 12209# 12211 12212# 12214 12215# 12217 12218# 12220 12221#
12223 12224# 12226 12227# 12229 12230# 12232 12233# 12235 12236# 12238 12239# 12241
12242# 12244 12245# 12247 12248# 12250 12251# 12253 12254# 12256 12258# 12260 12262#
12264 12266# 12270 12283# 12285 12286# 12288 12289# 12291 12292# 12294 12295# 12297
12298# 12300 12302# 12304 12308# 12310 12315# 12317 12321# 12323 12327# 12329 12330#
12332 12333# 12336 12342# 12344 12364# 12356 12357# 12359 12360# 12362 12363# 12365
12366# 12368 12369# 12371 12373# 12375 12376# 12381 12386# 12388 12392# 12394 12398#
12400 12413# 12415 12440# 12450 12451# 12453 12454# 12456 12457# 12459 12460# 12462
12463# 12466# 12468# 12469# 12472# 12474 12475# 12477 12478# 12480 12481# 12483 12484#
12486 12487# 12489 12490# 12492 12493# 12495 12496# 12498 12499# 12501 12502# 12504
12505# 12507 12508# 12510 12511# 12513 12514# 12516 12517# 12519 12520# 12522 12523#
12525 12526# 12528 12529# 12531 12532# 12534 12535# 12537 12538# 12540 12541# 12543
12544# 12546 12547# 12549 12550# 12552 12553# 12555 12556# 12558 12559# 12561 12562#
12564 12565# 12567 12568# 12570 12571# 12573 12575# 12577 12579# 12581 12585# 12587
12600# 12602 12603# 12605 12606# 12608 12609# 12611 12612# 12614 12615# 12617 12619#
12621 12625# 12627 12632# 12634 12638# 12640 12644# 12646 12647# 12649 12650# 12652
12659# 12661 12671# 12673 12674# 12676 12677# 12679 12680# 12682 12683# 12685 12686#
12688 12690# 12692 12696# 12698 12703# 12705 12709# 12711 12715# 12717 12730# 12732
12765# 12767 12768# 12770 12771# 12773 12774# 12776 12777# 12779 12780# 12782 12783#
12785 12788# 12791 12792# 12794 12795# 12797 12798# 12800 12801# 12803 12804# 12806
12807# 12809 12810# 12812 12813# 12815 12816# 12818 12819# 12821 12822# 12824 12825#
12827 12828# 12830 12831# 12833 12834# 12836 12837# 12839 12840# 12842 12843# 12845
12846# 12848 12849# 12851 12852# 12854 12855# 12857 12858# 12860 12861# 12863 12864#



```
                    14656  14664#  14666  14670#  14701  14702#  14705#  14707  14708#  14710  14711#  14713
                    14714#  14716  14717#  14721  14723#  14725  14727#  14728  14732#  14734  14738#  14740  14747#
                    14749  14750#  14752  14758#  14760  14764#  14786  14815#  14817  14818#  14820  14821#  14823
                    14824#  14826  14830#  14838  14843#  14846  14850#  14861  14866#  14867  14871  14873  14876
                    14877  14879
                    14900#  14902  14903#  14906  14909#  14911  14915#  14917  14921#  14923  14930#  14932  14936#
                    14938  14939#  14941  14942#  14944  14945#  14957  14958#  14960  14961#  14963  14964#  14972
                    14987#  14989  14993#  14995  14999#  15001  15005#  15007  15008#  15010  15011  15012  15013#
                    15015  15016#  15018  15019#  15021  15022#  15024  15025#  15027  15028#  15030  15034#  15036
                    15041#  15043  15047#  15049
$40CAT= ****** U    1046
.      = 037470     203#  205  208#  221  222#  224#  226#  228#  232#  236#  246#  294  380#
                    381#  382#  383#  392#  480#  486  487#  489#  491#  534#  536#  537#  538#
                    539#  541#  542#  543#  544#  546#  547#  548#  549#  551#  552#  553#  554#
                    556#  557#  558#  559#  561#  562#  563#  564#  566#  567#  568#  569#  571#
                    572#  573#  574#  576#  577#  578#  579#  581#  582#  583#  584#  586#  587#
                    588#  589#  591#  592#  593#  594#  596#  597#  598#  599#  601#  602#  603#
                    604#  606#  607#  608#  609#  611#  612#  613#  614#  775  785  892#  905
                    1010  1081  1082  1174  1226#  1236  1328#  1329  1335#  1389  1678  1707  1750#
                    1769#  1785#  1787#  1841  1844  1865  1954  2005  2088  2092  2138  2169  2202
                    2205  2281#  2478#  2675#  2872#  3070#  3267#  3520#  3717#  3914#  4111#  4308#  4505#
                    4702#  4899#  5097#  5274#  5481#  5688#  5885#  6102#  6309#  6516#  6723#  6930#  7137#
                    7343#  7517#  7760#  7957#  8137#  8294#  8455#  8668#  8834#  8960#  9277#  9594#  9911#
                    10228#  10545#  10862#  11179#  11496#  11813#  12130#  12447#  12764#  13081#  13398#  13713#  13856#
                    14480#  14698#  14800#  15310#  15312#
.ADC   = 000100      1#  5162  5232  8009  8097  8238  8398  8533  8617  8696  8783  8865  8916
                    12905  12973  13792  13868  14770  14773
.ADD   = 000000      1#  9095  9166  9231  9412  9483  9548  9729  9800  9865  10046  10117  10182
                    10363  10434  10499  10680  10686  10751  10754  10816  10997  11068  11133  11314  11385
                    11450  11631  11702  11767  11948  12019  12084  12265  12336  12401  12582  12653  12718
                    12899  12970  13035  13216  13287  13352  13533  13604  13669  14863  14933
.ADDWC= 000020       1#  11637  11705
.ADVAN  006050      1429#  1569
.AND   = 000260      1#  3092  3181  3309  3407  7861  9735  9803  14535  14836  14906  14970  14991
.B80   = 002000      1#  7410  7617  7644  7827  7988  8871  8922  14571  14854  15037
.B81   = 002400      1#  14577  14592  14824
.B84   = 003000      1#  13732  13760  13786  14559  14741
.B87   = 003400      1#  2362  2442  2659  2639  2756  2836  2953  3033  3160  3249  3386  3484
                    3601  3681  3798  3878  3995  4075  4192  4272  4389  4469  4586  4666  4783
                    4863  4980  5060  5168  5238  5360  5445  5567  5652  5774  5859  5981  6066
                    6188  6273  6395  6480  6602  6687  6809  6894  7016  7101  7223  7308  8232
                    8389  13798  14547  14735  14761  14848  14857  14918  14927  14982  14997  15031
.BC    = 001000      1#  7833  8250  8410  8545  8629  8705  8795  13983  14015  14342  14776
.BEGIN  003464      920#
.BN    = 000400      1#  2334  2340  2347  2353  2365  2414  2420  2427  2433  2450  2453  2531
                    2537  2544  2550  2562  2611  2617  2624  2630  2647  2650  2728  2734  2741
                    2747  2759  2808  2814  2821  2827  2844  2847  2905  2931  2938  2944  2956
                    3005  3011  3018  3024  3041  3044  3132  3138  3145  3151  3163  3221  3227
                    3234  3240  3257  3260  3358  3364  3371  3377  3388  3456  3462  3469  3475
                    3492  3498  3573  3579  3586  3592  3604  3653  3659  3666  3672  3689  3692
                    3770  3776  3783  3789  3801  3850  3856  3863  3869  3886  3889  3967  3973
                    3980  3986  3998  4047  4053  4060  4066  4083  4086  4164  4170  4177  4183
                    4195  4244  4250  4257  4263  4280  4283  4361  4367  4374  4380  4392  4441
                    4447  4454  4460  4477  4480  4558  4564  4571  4577  4589  4638  4644  4651
                    4657  4674  4677  4755  4761  4768  4774  4786  4835  4841  4848  4854  4871
                    4874  4952  4958  4965  4971  4983  5032  5038  5045  5051  5068  5071  5140
```

```
5143  5150  5153  5171  5210  5213  5220  5223  5246  5249  5332  5338  5345
5351  5363  5417  5423  5430  5436  5453  5456  5537  5542  5552  5558  5570
5624  5630  5637  5643  5660  5663  5746  5752  5759  5765  5777  5831  5837
5944  5950  5967  5970  5953  5959  5966  5972  5984  6038  6044  6051  6057
6074  6077  6160  6166  6173  6179  6191  6245  6251  6258  6264  6281  6284
6367  6373  6380  6386  6398  6452  6458  6465  6471  6488  6491  6574  6580
6587  6593  6605  6659  6665  6672  6678  6695  6698  6781  6787  6794  6800
6812  6866  6872  6878  6885  6902  6905  6988  6994  7001  7007  7019  7073
7079  7086  7092  7109  7112  7195  7201  7208  7214  7226  7280  7286  7293
7299  7316  7319  7444  7447  7478  7481  7488  7491  7681  7687  7721  7724
7731  7734  7836  7849  7852  7868  7871  7958  8012  8079  8082  8100  8107
8110  8204  8210  8217  8223  8260  8267  8270  8361  8367  8374  8380  8382
8420  8427  8430  8514  8517  8524  8527  8534  8598  8601  8604  8611  8632
8640  8643  8708  8758  8764  8771  8777  8790  8806  8809  8812  8819  8825
8838  8863  8932  9135  9141  9148  9154  9206  9212  9219  9222  9243  9246
9249  9252  9458  9465  9471  9523  9529  9536  9542  9556  9563  9566  9769
9775  9782  9788  9840  9846  9853  9859  9873  9880  9883  10086  10089  10099
10105  10157  10163  10170  10176  10190  10197  10200  10403  10409  10416  10422  10474
10480  10487  10493  10507  10514  10517  10720  10726  10733  10739  10791  10797  10804
10810  10824  10831  10834  11037  11043  11050  11056  11108  11114  11121  11127  11141
11146  11151  11354  11360  11367  11373  11430  11431  11438  11444  11458  11465  11468
11671  11677  11684  11690  11742  11748  11755  11761  11775  11782  11788  11795  11991
12001  12007  12059  12065  12072  12078  12085  12091  12102  12305  12311  12318  12324
12376  12382  12389  12395  12404  12410  12416  12422  12429  12435  12441  12448  12454
12706  12712  12726  12733  12736  12739  12745  12752  12758  13010  13016  13023  13029
13043  13050  13063  13069  13082  13089  13095  13327  13333  13340  13346  13360  13367
13370  13373  13379  13386  13392  13444  13450  13457  13463  13477  13484  13487  13745
13748  13763  13777  13780  13801  13815  13818  13825  13828  13986  13993  13999  14018
14024  14040  14056  14066  14076  14086  14096  14106  14116  14128  14138  14148  14158
14168  14178  14188  14198  14214  14233  14246  14259  14272  14285  14298  14311  14324
14337  14363  14386  14404  14411  14414  14451  14455  14586  14601  14614  14657
14660  14667  14670  14764  14779  14787  14875  14945  15000  15044  15050
:BSBRG= 160000  1#  14756  14826  14916  15001
:BSIMM= 100000  1#  2310  2335  2341  2348  2354  2363  2366  2390  2415  2421  2428  2434
2443  2451  2454  2507  2526  2551  2704  2729  2736  2743  2757  2762  2768  2774
2809  2815  2822  2828  2837  2915  2930  2934  2940  2947  2953  2959  2965  2971
2987  2981  3006  3012  3019  3022  3034  3040  3047  3108  3114  3120  3126  3133
3152  3161  3164  3197  3322  3328  3334  3340  3347  3353  3359  3365  3371  3377
3365  3372  3378  3387  3390  3396  3402  3409  3415  3421  3427  3433  3439  3446
3543  3574  3580  3587  3593  3602  3608  3614  3621  3627  3633  3639  3645  3652
3690  3693  3746  3771  3777  3784  3790  3796  3802  3809  3815  3821  3827  3834
3870  3879  3887  3890  3943  3946  3952  3958  3965  3971  3977  3983  3989  3996
4054  4061  4067  4076  4084  4087  4140  4146  4153  4159  4171  4178  4184  4193
4220  4245  4251  4258  4264  4273  4281  4287  4293  4299  4337  4362  4368  4375
4390  4393  4417  4442  4448  4455  4461  4467  4470  4474  4478  4481  4524  4559
4572  4578  4587  4590  4614  4626  4636  4643  4650  4654  4658  4667  4671  4675
4756  4762  4769  4775  4784  4787  4811  4817  4824  4830  4834  4837  4841  4844
4875  4928  4953  4959  4966  4972  4981  4987  4994  5000  5033  5039  5043  5046
5061  5069  5072  5125  5141  5144  5151  5157  5164  5169  5172  5176  5196  5211
5221  5224  5239  5247  5250  5305  5333  5339  5346  5352  5356  5361  5364  5369
5418  5424  5431  5437  5446  5454  5457  5512  5540  5664  5719  5747  5753  5760
5571  5597  5625  5631  5638  5644  5653  5661  5747  5753  5760  5766  5774  5780
5766  5775  5778  5804  5832  5838  5845  5851  5860  5868  5871  5877  5885  5892
5960  5967  5973  5982  5985  6011  6039  6045  6052  6058  6067  6075  6078  6084
6133  6161  6167  6174  6180  6189  6192  6218  6246  6252  6259  6265  6274
```

```
                6282  6285  6340  6368  6374  6381  6387  6396  6399  6425  6453  6459  6466
                6472  6481  6485  6498  6547  6575  6581  6588  6594  6603  6606  6632  6660
                6666  6673  6677  6680  6686  6689  6754  6782  6788  6795  6801  6810  6813
                6839  6862  6873  6880  6886  6896  6903  6906  6961  6989  6995  7002  7008
                7017  7020  7046  7074  7080  7087  7093  7102  7110  7113  7168  7196  7202
                7209  7215  7224  7267  7283  7291  7287  7294  7300  7309  7317  7320  7411
                7423  7445  7468  7467  7478  7482  7489  7442  7618  7645  7660  7682  7685
                7700  7723  7725  7736  7736  7850  7837  7837  7850  7863  7869  7872  7956
                7959  7989  8001  8013  8050  8080  8083  8089  8101  8108  8111  8180  8205
                8211  8218  8224  8233  8251  8261  8268  8271  8337  8362  8368  8375  8381
                8340  8343  8411  8421  8428  8431  8467  8515  8518  8525  8528  8546  8549
                8577  8593  8602  8609  8612  8630  8633  8641  8644  8706  8709  8737  8759
                8765  8772  8778  8796  8745  8807  8810  8872  8886  8889  8896  8899  8923
                8930  8933  9111  9136  9142  9149  9155  9182  9207  9213  9220  9226  9237
                9240  9247  9250  9454  9452  9459  9466  9472  9499  9524  9530  9537  9543
                9554  9557  9564  9567  9745  9770  9776  9783  9789  9816  9841  9847  9854
                9860  9871  9874  9881  9894  10062  10087  10093  10100  10106  10133  10158  10164
                10171  10177  10188  10191  10198  10201  10379  10404  10410  10417  10423  10450  10475
                10481  10488  10491  10505  10509  10515  10518  10636  10721  10727  10734  10740  10767
                10742  10788  10805  10811  10829  10835  10832  10835  11013  11038  11044  11051  11057
                11084  11109  11115  11132  11138  11139  11142  11149  11152  11330  11355  11361  11368
                11374  11401  11426  11430  11438  11445  11456  11459  11466  11469  11647  11672  11678
                11665  11691  11718  11743  11749  11756  11762  11773  11776  11783  11786  11964  11989
                11985  12002  12008  12015  12060  12066  12073  12079  12090  12093  12100  12103  12281
                12306  12312  12319  12368  12377  12383  12390  12396  12407  12410  12417  12420
                12588  12623  12629  12636  12642  12669  12694  12700  12707  12713  12724  12727  12734
                12737  12915  12940  12946  12953  12963  12970  13011  13017  13024  13030  13041  13044
                13051  13054  13232  13257  13263  13276  13303  13328  13334  13341  13347  13358
                13361  13368  13371  13549  13574  13580  13587  13593  13620  13645  13651  13658  13664
                13675  13678  13686  13688  13733  13746  13749  13761  13754  13778  13781  13787  13799
                13802  13816  13819  13826  13829  13984  13987  13994  14000  14007  14016  14019  14025
                14041  14057  14067  14077  14087  14097  14107  14117  14129  14139  14149  14159  14169
                14179  14189  14199  14215  14234  14247  14260  14273  14286  14299  14312  14325  14338
                14365  14384  14387  14393  14396  14402  14405  14412  14415  14430  14452  14548  14560
                14572  14578  14587  14593  14602  14617  14620  14636  14658  14661  14668  14671  14736
                14742  14762  14765  14777  14780  14788  14849  14855  14858  14919  14925  14928  14983
                14998  15032  15038  15045  15051
.BSMEM= 140000    1#
.BZ   = 001400    1#  2309  2389  2506  2586  2703  2783  2900  2980  3107  3196  3333  3431
                3548  3628  3745  3825  3942  4022  4139  4219  4336  4416  4533  4613  4730
                4810  4927  5007  5124  5194  5304  5389  5511  5596  5718  5803  5925  6010
                6132  6217  6339  6424  6546  6631  6753  6838  6960  7045  7167  7252  7422
                7459  7559  7638  7965  8000  8057  8088  8179  8336  8486  8576  8736  9110
                9181  9236  9427  9498  9553  9744  9815  9870  10061  10132  10187  10378  10449
                10504  10696  10766  10821  11012  11083  11138  11329  11400  11455  11646  11717  11772
                11963  12034  12089  12280  12351  12406  12597  12668  12723  12914  12965  13040  13231
                13302  13357  13548  13619  13674  14006  14364  14401  14429  14616  14635
.CNVRT  006160  1462#  1568
.CONVR  006154  1461#  1567
.CO   = 000400    1#
.DATAC  007420  1564  1741#
.DBR  = 000400    1#  2283  2290  2306  2331  2337  2344  2350  2356  2370  2386  2411  2417
                2424  2430  2436  2447  2480  2487  2503  2528  2534  2541  2547  2553  2567
                2583  2608  2614  2621  2627  2633  2644  2677  2684  2700  2725  2731  2738
                2744  2750  2764  2780  2805  2811  2818  2824  2830  2841  2874  2881  2897
                2922  2928  2935  2941  2947  2961  2977  3002  3008  3015  3021  3027  3038
```

3072	3079	3092	3104	3129	3135	3142	3148	3154	3168	3181	3193	3218
3224	3231	3247	3263	3282	3288	3295	3307	3314	3328	3336	3361	3366
3374	3380	3394	3407	3422	3428	3433	3459	3466	3472	3478	3489	3522
3529	3535	3570	3576	3583	3589	3596	3603	3609	3616	3628	3663	3669
3675	3681	3719	3726	3733	3739	3746	3753	3773	3780	3787	3825	3847
3853	3860	3866	3872	3883	3889	3896	3916	3923	3930	3937	3983	3989
4003	4019	4041	4060	4067	4073	4080	4087	4093	4100	4113	4161	4167
4174	4180	4196	4213	4220	4226	4233	4240	4247	4254	4267	4310	4317
4333	4339	4354	4371	4377	4383	4390	4397	4403	4413	4451	4457	4463
4474	4507	4514	4530	4537	4543	4550	4557	4563	4572	4610	4636	4641
4648	4654	4671	4687	4694	4700	4707	4714	4727	4734	4771	4777	4791
4807	4862	4875	4891	4898	4904	4911	4918	4924	4931	4938	4955	4962
4968	4974	4988	5004	5011	5017	5024	5031	5037	5044	5051	5112	5137
5147	5155	5168	5184	5191	5197	5204	5211	5217	5224	5310	5316	5322
5328	5334	5348	5364	5371	5377	5384	5391	5397	5404	5414	5420	5427
5433	5439	5454	5470	5477	5483	5490	5497	5503	5510	5542	5549	5555
5561	5575	5588	5604	5611	5617	5624	5631	5637	5644	5646	5657	5690
5697	5715	5728	5744	5751	5757	5764	5771	5777	5784	5792	5800	5809
5815	5821	5834	5850	5857	5863	5870	5877	5883	5890	5922	5931	5937
5943	5960	5974	5990	5997	6003	6010	6017	6023	6030	6038	6035	6041
6048	6054	6071	6087	6094	6100	6107	6114	6121	6127	6016	6163	6170
6176	6182	6196	6214	6221	6227	6234	6241	6247	6254	6144	6261	6278
6311	6318	6336	6345	6351	6357	6364	6371	6377	6384	6150	6389	6421
6430	6436	6442	6445	6452	6458	6465	6472	6478	6485	6157	6525	6552
6558	6564	6571	6577	6584	6590	6597	6604	6610	6617	6624	6643	6650
6662	6669	6675	6681	6688	6694	6701	6708	6714	6721	6728	6765	6771
6791	6797	6803	6817	6824	6830	6837	6844	6850	6857	6863	6869	6876
6899	6932	6939	6957	6964	6970	6977	6984	6990	6997	7004	7010	7024
7042	7051	7057	7063	7070	7076	7083	7089	7095	7106	7139	7146	7164
7173	7179	7185	7192	7198	7205	7211	7217	7231	7249	7258	7264	7270
7277	7283	7290	7296	7302	7313	7344	7345	7377	7383	7389	7395	7401
7407	7413	7419	7428	7434	7441	7450	7456	7475	7485	7518	7554	7557
7569	7575	7581	7587	7593	7599	7605	7614	7623	7629	7635	7641	7647
7656	7662	7678	7687	7696	7705	7718	7728	7761	7797	7803	7810	7824
7846	7855	7861	7865	7904	7925	7931	7937	7967	7973	7979	7985	8015
8063	8076	8104	8139	8145	8176	8188	8191	8201	8207	8215	8220	8233
8244	8256	8302	8333	8345	8351	8358	8364	8371	8377	8383	8421	8456
8477	8483	8492	8498	8504	8511	8521	8534	8542	8552	8567	8573	8582
8588	8595	8605	8620	8626	8637	8669	8687	8702	8712	8727	8733	8742
8748	8755	8768	8786	8792	8803	8835	8868	8882	8892	8919	8926	8967
9092	9095	9119	9125	9132	9145	9163	9166	9180	9196	9203	9216	9228
9243	9284	9409	9412	9436	9442	9449	9462	9480	9483	9507	9513	9520
9533	9545	9560	9601	9608	9628	9653	9659	9674	9693	9717	9800	9824
9830	9837	9850	9862	9877	9918	10043	10046	10070	10076	10083	10096	10114
10117	10141	10147	10154	10167	10179	10194	10235	10360	10363	10387	10393	10400
10413	10431	10434	10458	10464	10471	10484	10496	10511	10552	10677	10680	10704
10710	10717	10730	10748	10751	10775	10781	10788	10801	10813	10828	10869	10994
10997	11021	11027	11034	11047	11065	11068	11082	11088	11105	11118	11130	11145
11186	11311	11314	11338	11344	11351	11364	11382	11385	11469	11415	11422	11435
11447	11462	11503	11628	11631	11655	11661	11668	11681	11693	11702	11726	11732
11739	11752	11764	11779	11820	11945	11948	11972	11978	11985	11998	12016	12019
12043	12049	12056	12069	12081	12086	12137	12262	12265	12289	12295	12302	12315
12333	12336	12360	12366	12373	12386	12398	12413	12454	12574	12582	12606	12612
12619	12632	12650	12653	12677	12683	12690	12703	12715	12730	12771	12896	12899
12923	12929	12936	12949	12967	12970	12994	13000	13007	13020	13032	13047	13088
13213	13216	13240	13246	13253	13266	13284	13287	13311	13317	13324	13337	13349

CROSS REFERENCE TABLE -- USER SYMBOLS

13364 13405 13520 13533 13557 13563 13570 13583 13601 13604 13628 13634 13641
13654 13666 13701 13714 13729 13742 13751 13757 13774 13783 13795 13812 13822
13857 13874 13900 13910 13926 14001 14027 14063 14063 14073 14083 14093 14103
14113 14125 14136 14145 14160 14164 14176 14186 14186 14204 14211 14217 14223
14240 14256 14269 14284 14300 14306 14323 14340 14360 14360 14295 14301 14308
14314 14321 14327 14334 14350 14361 14373 14380 14408 14420 14426 14435 14441
14448 14481 14511 14517 14536 14541 14544 14560 14566 14562 14568 14574 14589
14607 14613 14626 14632 14641 14647 14654 14664 14698 14714 14723 14732 14738
14747 14758 14784 14821 14833 14839 14845 14851 14860 14866 14894 14903 14909
14915 14921 14930 14936 14956 14967 14973 14979 14985 14994 15010 15016 15022

.DBRSH= 001400

1# 2360 2440 2567 2637 2754 2834 2951 3031 3158 3247 3384 3482
3599 3679 3758 3876 3943 4073 4190 4270 4387 4467 4584 4664 4781
4861 4978 5058 5208 5243 5363 5460 5572 5657 5979 6064 6186 6271
6343 6478 6600 6685 6807 6892 7014 7099 7221 7306 7372 7564 7950
8230 8387

.DBRSP= 003400

1# 9101 9169 9418 9486 9735 9803 10052 10120 10369 10437 10686 10754
11003 11071 11320 11388 11637 11705 11954 12022 12271 12339 12588 12656 12905
12973 13222 13290 13539 13607

.DD = 003000

2306# 2386# 2503# 2583# 2700# 2780# 2897# 2977# 3092# 3104# 3181# 3193# 3309#
3324# 3330# 3407# 3422# 3438# 3515# 3525# 3742# 3822# 3938# 4019# 4136# 4216#
4333# 4413# 4530# 4610# 4727# 4807# 4924# 5004# 5121# 5191# 5301# 5386# 5508#
5593# 5715# 5800# 5922# 6007# 6129# 6214# 6336# 6421# 6543# 6628# 6750# 6835#
6957# 7042# 7164# 7249# 7419# 7485# 7534# 7611# 7656# 7696# 7815# 7861# 8054#
8176# 8247# 8333# 8407# 8493# 8565# 8573# 8645# 8702# 8733# 8792# 9095# 9096
9107# 9166# 9167# 9176# 9231# 9412# 9413# 9418# 9424# 9483# 9484# 9486# 9495#
9548# 9729# 9730# 9735# 9741# 9800# 9801# 9803# 9812# 9865# 10046# 10047 10052#
10058# 10117# 10118 10120# 10125# 10182# 10363# 10364 10369# 10375# 10434# 10435 10437#
10446# 10499# 10680# 10681 10686# 10742# 10751# 10752 10754# 10763# 10816# 10997# 10998
11009# 11068# 11069 11080# 11133# 11314# 11315 11320# 11326# 11365# 11386 11388# 11397#
11450# 11631# 11632 11637# 11643# 11702# 11703 11705# 11714# 11767# 11948# 11949 11954#
11960# 12019# 12020 12022# 12031# 12084# 12085 12271# 12277# 12336# 12337 12340# 12401#
12582# 12583 12594# 12653# 12654 12665# 12716# 12900 12900 12911# 12970# 12971 12982#
13035# 13216# 13217 13222# 13230# 13287# 13288 13340# 13349# 13352# 13533# 13534 13545#
13604# 13605 13616# 13665# 13680# 14361# 14426# 14536# 14541# 14613# 14632# 14720# 14729#
14753# 14827# 14836# 14842# 14863# 14906# 14912# 14933# 14970# 14976# 14991#

.DEC = 000160

1# 7952 7997 8085 8901 8913 8982 9233 9299 9550 9616 9867 9933
10184 10250 10501 10567 10818 10884 11135 11201 11452 11518 11769 11835 12085
12152 12403 12469 12720 12786 13037 13103 13354 13420 13539 13607 13671 14003
14037 14398

.DELAY 007304
.DMEM = 002400

1562 1714#
1# 2318 2324 2352 2404 2515 2521 2595 2601 2712 2718 2792 2798
2809 2815 2889 2935 3086 3116 3122 3175 3205 3211 3303 3318 3342
3348 3401 3416 3440 3446 3557 3563 3637 3643 3754 3760 3834 3840
3951 3957 4031 4037 4148 4154 4228 4234 4345 4351 4425 4431 4542
4548 4632 4638 4739 4745 4819 4825 4936 4942 5016 5022 5106 5121
5165 5176 5191 5236 5289 5365 5374 5380 5496 5502 5581 5587 5703
5709 5788 5794 5910 5916 5992 6001 6117 6123 6202 6208 6324 6330
6409 6415 6531 6537 6616 6622 6738 6744 6823 6829 6945 6951 7030
7036 7152 7158 7237 7243 7468 7671 7711 7809 7839 7946 7991 7994
8054 8069 8154 8164 8241 8247 8311 8321 8401 8407 8474 8690 8856
8875 8904 8990 8991 8994 8997 9000 9003 9006 9009 9012 9015 9018
9021 9024 9027 9030 9033 9036 9039 9042 9045 9048 9051 9054 9057
9060 9063 9066 9069 9072 9075 9078 9081 9107 9178 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

.DNOP = 000000 1#

.DOUT0= 002000 1#

.DOUT1= 001000 1#

```
                    4231  4237  4339  4342  4348  4364  4419  4422  4428  4434  4536  4539  4545
                    4551  4616  4619  4626  4631  4733  4736  4742  4748  4813  4816  4822  4828
                    4884  4933  4936  5010  5013  5015  5028  5127  5130  5133  5197  5200
                    5303  5307  5313  5319  5326  5327  5330  5410  5514  5520  5526  5532
                    5537  5606  5611  5617  5621  5723  5729  5806  5812  5818  5824  5828
                    5944  5948  5953  6013  6016  6031  6136  6141  6147  6153  6220  6226
                    6332  6338  6343  6348  6352  6427  6433  6439  6445  6549  6555  6561
                    6567  6634  6648  6654  6660  6742  6748  6768  6774  6841  6847  6853  6859
                    6963  6969  6975  6981  7048  7050  7056  7066  7170  7176  7182  7188  7255
                    7261  7267  7273  7279  7404  7431  7437  7462  7465  7471  7548  7554
                    7611  7638  7663  7668  7674  7708  7714  7791  7815  7821  7e 2  7861
                    7922  7982  8021  8026  8066  8072  8108  8185  8191  8197  8339  8342  8348
                    8354  8409  8445  8501  8507  8567  8591  8734  8745  8751  8872  9113
                    9116  9122  9128  9184  9187  9193  9198  9430  9433  9439  9445  9501  9504
                    9510  9516  9747  9750  9756  9763  9818  9821  9827  9833  10064  10067  10073
                   10079  10135  10138  10144  10150  10381  10384  10390  10396  10452  10455  10461  10467
                   10698  10701  10707  10713  10769  10772  10778  10784  11015  11018  11024  11030  11086
                   11089  11095  11101  11332  11335  11341  11347  11403  11406  11412  11418  11649  11652
                   11658  11664  11720  11723  11726  11732  11966  11969  11975  11981  12037  12040  12046
                   12052  12283  12286  12292  12298  12354  12357  12363  12369  12600  12603  12609  12615
                   12671  12674  12680  12686  12817  12820  12826  12832  12888  12991  12997  13003  13324
                   13237  13243  13249  13305  13308  13314  13320  13551  13554  13560  13566  13622  13625
                   13631  13637  13738  13754  13770  13808  13876  13879  13882  13885  13888  13982  13985
                   14367  14370  14376  14432  14438  14444  14490  14493  14526  14541  14553  14565  14638
                   14644  14650  14720  14729  14753  14815  14827  14842  14872  14897  14912  14942  14961
.DSP  = 003000      14976  14991  15013  15019  15025
                      1#  2286  2297  2300  2377  2380  2483  2494  2497  2574  2577  2680  2691
                    2694  2771  2774  2877  2880  2891  2968  2971  3075  3083  3089  3098  3172
                    3178  3187  3292  3300  3306  3315  3321  3398  3404  3413  3419  3525  3536
                    3539  3616  3619  3722  3733  3736  3813  3816  3919  3930  3933  4010  4013
                    4116  4127  4130  4207  4210  4313  4324  4327  4404  4407  4510  4521  4524
                    4601  4604  4707  4718  4721  4798  4801  4904  4915  4918  4995  4998  5102
                    5109  5115  5160  5163  5179  5185  5230  5233  5279  5286  5292  5371  5377
                    5486  5493  5499  5578  5584  5633  5700  5706  5785  5791  5900  5907  5913
                    5982  5988  6107  6114  6120  6199  6205  6314  6321  6327  6406  6412  6521
                    6528  6534  6613  6619  6728  6735  6741  6820  6826  6935  6942  6948  7027
                    7033  7142  7149  7155  7234  7240  7247  7368  7374  7380  7386  7521  7551
                    7560  7566  7572  7578  7608  7611  7650  7690  7764  7767  7812  7815  7831
                    7858  7898  7901  7907  7928  7934  7963  7998  8004  8007  8010  8018  8024
                    8027  8030  8033  8042  8086  8092  8095  8098  8142  8157  8160  8167  8170
                    8236  8239  8298  8314  8317  8324  8327  8396  8399  8459  8468  8471  8531
                    8534  8561  8564  8615  8618  8672  8681  8684  8694  8697  8721  8724  8781
                    8784  8838  8847  8850  8863  8869  8963  8966  9002  9007  9011  9014  9017
                    9070  9073  9077  9079  9083  9085  9089  9098  9160  9172  9231  9234  9287
                    9290  9293  9296  9300  9302  9401  9415  9418  9477  9486  9489  9548  9551
                    9604  9607  9610  9613  9617  9619  9718  9732  9735  9794  9803  9806  9865
                    9868  9921  9924  9927  9930  9934  9936  10035  10049  10052  10111  10120  10123
                   10182  10185  10238  10241  10244  10247  10251  10253  10362  10366  10369  10428  10437
                   10440  10499  10502  10555  10558  10561  10564  10568  10570  10669  10683  10686  10745
                   10754  10757  10816  10819  10872  10875  10878  10881  10885  10887  10986  11000  11062
                   11074  11133  11136  11189  11192  11195  11198  11202  11204  11303  11317  11320  11379
                   11388  11391  11450  11453  11506  11509  11512  11515  11519  11521  11620  11634  11637
                   11696  11705  11708  11767  11770  11823  11826  11829  11832  11836  11838  11937  11951
                   11954  12013  12022  12025  12084  12087  12140  12143  12146  12149  12153  12155  12254
                   12268  12330  12342  12401  12404  12457  12460  12463  12466  12470  12472  12571  12585
                   12647  12659  12718  12721  12774  12777  12780  12783  12787  12789  12888  12902  12964
```

```
                    12976  13035  13038  13091  13094  13097  13100  13104  13106  13205  13219  13222  13281
                    13290  13293  13352  13355  13408  13411  13414  13417  13421  13423  13522  13536  13588
                    13610  13669  13672  13717  13720  13723  13726  13790  13793  13860  13863  13866  13927
                    13980  13983  13986  13989  13992  13995  13998  14051  14054  14057  14066  14069  14071
                    14004  14009  14013  14031  14034  14038  14046  14049  14059  14069  14079  14089  14099
                    14109  14121  14131  14141  14151  14161  14171  14181  14191  14207  14220  14226  14239
                    14253  14265  14278  14291  14304  14317  14330  14349  14353  14390  14399  14417  14508
                    14530  14538  14541  14580  14583  14595  14598  14604  14623  14702  14705  14708  14711
                    14717  14720  14726  14729  14750  14753  14766  14771  14774  14810  14813  14818  14824
                    14827  14830  14836  14842  14863  14869  14886  14889  14891  14900  14906  14912  14933
                    14939  14955  14964  14970  14976  14988  14991
.DO    = 000400      1#
.FO    = 000020      1#  2297   2315   2377   2395   2494   2512   2574   2592   2691   2709   2771   2789
                    2888   2906   2968   2986   3048   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
.INC   = 000060      1#  7830   8003   8006   8091   8094   8235   8395   8530   8614   8693   8780   8862
                    8910  12271  12339  13789  13965  14012  14389  14767  14809  14812  14885  14888
.LORN  = 000240      1#  9418   9486
.MINUS= 000360       1#
.MSTCL  007334      1561   1725#
.MO    = 004000      1#
.OR    = 000300      1#  3324   3422   7554   7611   7815  10052  10120  14541  14720  14729  14753  14827
                   14842  14912  14976
.PLUS  = 000000      1#
.RESOS  006122      1450#  1560
.ROMCL  007352      1563   1730#
.SAVOS  006062      1436#  1559
.SBR   = 060000      1#  2286   2294   2300   2306   2312   2337   2350   2356   2360   2374   2380   2386
                    2392   2417   2430   2436   2440   2463   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   3146   3154
                    3158   3172   3181   3184   3193   3199   3224   3237   3243   3247   3282   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   3798   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   4146   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   4982   4998   5004   5010   5035   5046   5054   5058   5102   5115
                    5130   5159   5162   5165   5185   5200   5229   5232   5235   5379   5386   5389   5395
                    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
```

F11

5532	5565	5578	5581	5587	5593	5599	5602	5605	5611	5617	5650	5693
5700	5703	5709	5715	5721	5724	5727	5733	5739	5772	5786	5788	5794
5800	5806	5809	5812	5818	5824	5857	5900	5907	5910	5916	5922	5928
5931	5934	5940	5946	5979	5982	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	6521	6528	6531	6537	6543	6549	6552	6555
6561	6567	6600	6613	6616	6622	6628	6634	6637	6640	6646	6652	6685
6728	6736	6738	6744	6750	6756	6759	6762	6768	6774	6807	6820	6823
6829	6836	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	7696	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	7957	7970	7973	7976
7982	7997	8003	8006	8009	8021	8024	8027	8030	8045	8048	8063	8066
8085	8091	8094	8097	8142	8164	8170	8176	8182	8191	8194	8197	3207
8220	8226	8230	8236	8240	8254	8257	8299	8321	8327	8333	8339	8348
8351	8354	8364	8377	8383	8387	8395	8398	8414	8417	8459	8468	8471
8474	8483	8492	8495	8501	8504	8507	8530	8533	8542	8561	8564	8573
8582	8585	8591	8614	8617	8626	8672	8681	8684	8690	8693	8696	8702
8721	8724	8733	8743	8745	8751	8761	8774	8780	8783	8792	8838	8847
8850	8853	8862	8865	8868	8901	8910	8913	8916	8919	8970	8973	8976
8979	8982	8986	9035	9101	9116	9122	9128	9130	9151	9157	9166	9169
9187	9193	9199	9205	9222	9231	9233	9237	9240	9243	9296	9299	9302
9412	9433	9436	9439	9455	9468	9474	9483	9504	9510	9516	9536	9539
9548	9550	9604	9607	9610	9613	9616	9619	9729	9750	9756	9763	9772
9785	9791	9800	9821	9827	9833	9843	9861	9865	9867	9921	9924	9927
9930	9933	9936	10046	10067	10073	10079	10089	10105	10108	10117	10138	10144
10150	10160	10173	10182	10184	10238	10241	10244	10247	10250	10253	10363	10384
10390	10396	10406	10419	10435	10439	10455	10461	10467	10477	10480	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	11096
11101	11111	11124	11133	11135	11168	11192	11201	11204	11314	11335
11341	11347	11357	11370	11376	11385	11406	11412	11418	11428	11441	11450	11452
11506	11509	11512	11515	11518	11521	11631	11652	11658	11664	11674	11687	11693
11702	11723	11726	11735	11745	11748	11767	11769	11823	11826	11829	11832	11835
11838	11948	11969	11975	11981	11991	12004	12010	12019	12048	12046	12052	12062
12075	12084	12086	12140	12143	12146	12149	12152	12155	12265	12271	12286	12292
12298	12308	12321	12327	12336	12339	12367	12373	12376	12379	12392	12401	12403
12457	12460	12463	12466	12469	12472	12582	12585	12603	12608	12615	12653	12638
12644	12653	12656	12674	12680	12686	12696	12699	12709	12718	12720	12774	12777	12780
12783	12786	12789	12899	12905	12920	12926	12932	12942	12945	12961	12970	12973
12991	12997	13003	13013	13026	13035	13037	13041	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

```
                    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   14262   14265   14275   14278   14288   14291   14301
                    14304   14314   14317   14327   14330   14349   14361   14370   14376   14389   14398   14417   14420
                    14426   14432   14435   14438   14444   14490   14493   14496   14499   14502   14505   14508   14514
                    14520   14535   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
.SCOP1  004360   1086#   1558
.SELA = 000200      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   3154   3224   3237   3243   3361   3374   3380   3459   3472   3478   3576   3589
                    3595   3656   3669   3675   3773   3786   3792   3853   3866   3872   3970   3983   3989
                    4050   4063   4069   4167   4180   4186   4247   4260   4266   4364   4377   4383   4444
                    4457   4463   4561   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   5602   5703   5724   5788   5809   5910   5931   5985   6016   6117
                    6138   6202   6223   6324   6345   6409   6430   6531   6552   6616   6637   6738   6759
                    6823   6844   6945   6966   7030   7051   7152   7173   7237   7258   7389   7395   7428
                    7465   7581   7587   7593   7599   7663   7669   7665   7705   7967   7973   8045   8048
                    8063   8194   8207   8220   8226   8254   8257   8351   8364   8377   8383   8414   8417
                    8474   8492   8504   8582   8620   8742   8761   8774   8868   8919   9101   9138   9151
                    9157   9169   9209   9223   9455   9468   9474   9526   9539   9772   9785   9791   9843
                    9856  10089  10102  10108  10160  10173  10406  10419  10425  10477  10490  10723  10736
                   10742  10794  10907  11040  11053  11059  11111  11124  11357  11370  11376  11428  11441
                   11674  11687  11693  11745  11758  11951  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  14262  14275  14288  14301  14314  14327  14420  14435  14456
                   14607  14626  14641  14860  14876  14930  14996  15001
.SELB = 000220      1#   2286   2294   2300   2312   2321   2367   2374   2380   2392   2401   2407
                    2440   2463   2491   2497   2509   2518   2564   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   3096   3110   3119   3125   3158   3172   3178   3184   3195
                    3208   3214   3247   3262   3300   3306   3313   3321   3336   3345   3351   3364   3398
                    3404   3410   3419   3434   3443   3449   3456   3463   3533   3539   3551   3560   3566
                    3599   3613   3619   3631   3645   3646   3673   3720   3730   3736   3748   3757   3763
                    3796   3810   3816   3828   3837   3843   3876   3916   3927   3933   3945   3954   3960
                    3993   4007   4013   4035   4034   4040   4073   4116   4124   4130   4142   4151   4157
                    4190   4204   4210   4222   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   4822   4828   4861   4904   4912   4918   4930   4939   4945
                    4978   4992   4998   5010   5019   5025   5058   5102   5109   5112   5115   5127   5130
                    5156   5179   5182   5185   5197   5200   5226   5279   5286   5292   5295   5307   5313
                    5319   5325   5335   5348   5354   5358   5371   5377   5380   5392   5398   5404   5410
                    5420   5433   5439   5443   5466   5469   5494   5502   5514   5520   5526   5532   5542
                    5555   5561   5565   5578   5584   5587   5599   5605   5611   5617   5627   5640   5646
                    5650   5693   5700   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   5956   5969   5975   5979   5988   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
```

```
                    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  6869  6882  6888  6893  6935  6942  6948  6951
                    6963  6969  6975  6981  6991  7004  7010  7014  7027  7033  7036  7048  7054
                    7060  7066  7076  7089  7095  7099  7142  7149  7155  7158  7170  7176  7182
                    7188  7198  7211  7217  7221  7234  7240  7243  7255  7261  7267  7273  7283
                    7296  7302  7306  7347  7350  7353  7356  7359  7362  7368  7372  7374  7380
                    7386  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  7626  7632  7638  7663  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  8060  8066  8072  8142  8157  8160  8164  8167
                    8170  8182  8186  8191  8197  8230  8299  8314  8317  8321  8324  8327  8339
                    8342  8348  8354  8387  8459  8468  8471  8477  8489  8495  8501  8507  8561
                    8564  8567  8579  8586  8591  8672  8681  8684  8721  8724  8727  8738  8745
                    8751  8838  8847  8850  8853  8859  8878  8907  8970  8973  8976  8979  8985
                    9098  9113  9116  9122  9128  9160  9172  9184  9187  9193  9199  9287  9290
                    9293  9296  9302  9415  9430  9433  9439  9445  9477  9489  9501  9504  9510
                    9516  9604  9607  9610  9613  9619  9732  9747  9750  9756  9762  9794  9806
                    9818  9821  9827  9833  9921  9924  9927  9930  9936  10049 10064 10067 10073
                   10079 10111 10123 10135 10138 10144 10150 10238 10241 10244 10247 10253 10366
                   10381 10384 10390 10396 10428 10440 10452 10455 10461 10467 10555 10558 10561
                   10564 10570 10683 10698 10701 10707 10713 10745 10757 10769 10772 10778 10784
                   10872 10875 10878 10881 10887 11000 11015 11018 11024 11030 11062 11074 11086
                   11089 11095 11101 11109 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 11964 11969 11975 11981 12013 12025 12037 12040 12046 12052 12140
                   12143 12146 12149 12155 12260 12263 12266 12269 12272 12330 12342 12354 12357
                   12363 12369 12457 12460 12463 12466 12469 12472 12600 12603 12609 12615 12647
                   12659 12671 12674 12677 12680 12683 12774 12777 12780 12783 12789 12902 12917 12920
                   12926 12932 12964 12967 12970 12973 12991 12997 13003 13091 13094 13097 13100 13106
                   13219 13234 13237 13240 13243 13246 13281 13293 13305 13308 13314 13320 13408 13411
                   13414 13417 13463 13466 13469 13551 13554 13560 13566 13598 13610 13622 13625 13631
                   13637 13717 13720 13723 13726 13730 13754 13770 13808 13860 13863 13866 13873
                   13879 13882 13906 13909 13912 13915 13918
                   13921 13927 13930 13933 13936 13939 13942 13945 13948 13951 13954 13957 13962
                   13971 14009 14046 14207 14220 14226 14239 14252 14265 14278 14291 14304 14317
                   14330 14349 14367 14370 14376 14417 14432 14438 14444 14490 14493 14496
                   14499 14502 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 15023
.SIMM = 000000      1#  2283  2290  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
```

```
4555  4568  4584  4622  4628  4635  4648  4671  4704  4711  4739  4745  4752
4765  4791  4819  4825  4832  4845  4868  4901  4908  4936  4942  4949  4962
5000  5016  5022  5029  5042  5065  5099  5106  5133  5137  5147  5176  5203
5207  5217  5243  5262  5279  5316  5422  5428  5430  5436  5460  5461  5474
5487  5490  5493  5496  5523  5529  5536  5554  5575  5608  5614  5621  5634
5657  5680  5697  5730  5736  5743  5756  5782  5815  5821  5828  5841  5864
5897  5904  5937  5943  5950  5963  5989  6022  6028  6035  6048  6071  6104
6111  6144  6150  6157  6170  6196  6228  6236  6242  6255  6278  6311  6318
6351  6357  6364  6377  6403  6436  6442  6449  6462  6485  6518  6525  6558
6564  6571  6584  6610  6643  6649  6656  6669  6682  6705  6732  6765  6771
6778  6791  6817  6850  6856  6863  6876  6890  6932  6939  6972  6978  6985
6998  7024  7057  7063  7070  7083  7106  7130  7146  7179  7185  7192  7205
7231  7264  7270  7277  7290  7313  7344  7365  7377  7383  7401  7434  7441
7468  7475  7485  7518  7557  7569  7575  7605  7635  7671  7678  7711  7718
7728  7761  7797  7803  7809  7818  7839  7846  7855  7865  7898  7901  7904
7925  7931  7937  7940  7943  7961  7964  7979  8015  8018  8033  8036  8039
8059  8076  8104  8139  8145  8148  8151  8154  8188  8201  8214  8241  8264
8286  8302  8305  8308  8311  8345  8368  8371  8401  8424  8454  8457  8465
8498  8511  8521  8536  8552  8565  8568  8588  8595  8605  8620  8637  8669
8675  8678  8687  8712  8715  8718  8746  8755  8758  8786  8806  8836  8841
8844  8865  8875  8882  8902  8904  8926  8961  8964  8967  8991  8994  8997
8997  9000  9003  9006  9009  9012  9015  9018  9021  9024  9027  9030  9033
9036  9039  9042  9045  9048  9051  9054  9057  9060  9063  9066  9069  9072
9075  9078  9081  9084  9088  9092  9119  9125  9132  9145  9163  9190  9196
9203  9216  9228  9243  9278  9281  9284  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  9436  9442  9449  9462  9480  9507  9513  9520  9533  9545
9560  9595  9598  9601  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
9753  9759  9766  9779  9797  9824  9830  9837  9850  9856  9877  9912  9915
9918  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  10070  10076  10083
10096  10114  10141  10147  10154  10167  10176  10194  10224  10232  10235  10256  10259
10262  10265  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  10387  10393  10408  10413  10431  10458
10464  10471  10484  10496  10511  10546  10549  10552  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  10673  10677  10704  10710  10717  10730  10738  10775  10781  10788  10801
10813  10828  10863  10866  10869  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  10990
10994  11021  11027  11034  11047  11065  11092  11098  11105  11118  11130  11145  11180
11183  11186  11207  11210  11213  11216  11219  11222  11225  11228  11231  11234  11237
11240  11243  11246  11249  11252  11255  11258  11261  11264  11267  11270  11273  11277
11279  11282  11285  11288  11291  11294  11297  11300  11303  11307  11311  11338  11344
11351  11364  11382  11409  11415  11422  11435  11447  11462  11497  11500  11503  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  11624  11628  11655  11661  11668  11681  11699
11726  11732  11739  11752  11764  11779  11814  11817  11820  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 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 12262 12269 12295 12302 12315 12333 12360 12366 12373 12386 12398 12413
12448 12451 12454 12475 12478 12481 12484 12487 12490 12493 12496 12499 12502
12505 12508 12511 12514 12517 12520 12523 12526 12529 12532 12536 12538 12541
12544 12547 12550 12553 12556 12559 12562 12565 12568 12571 12575 12579 12606
12612 12619 12632 12650 12677 12683 12690 12703 12715 12730 12766 12768 12771
12782 12785 12788 12801 12804 12807 12810 12813 12816 12819 12822 12826 12828
12831 12834 12837 12840 12843 12846 12849 12852 12855 12858 12861 12864 12867
12870 12873 12876 12879 12882 12885 12888 12892 12896 12923 12928 12936 12949
12967 12994 13000 13007 13020 13032 13047 13062 13085 13088 13104 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 13345 13364 13399 13402 13405 13426 13429 13432 13435 13438 13441
13444 13447 13450 13453 13456 13459 13462 13465 13468 13471 13474 13477 13480
13483 13486 13489 13492 13495 13498 13501 13504 13507 13510 13513 13516 13519
13522 13526 13530 13557 13563 13570 13583 13601 13628 13634 13641 13654 13666
13681 13714 13736 13742 13751 13767 13774 13805 13812 13822 13857 13870 13873
13974 14021 14027 14031 14034 14053 14063 14073 14083 14093 14103 14113 14125
14135 14145 14155 14165 14175 14185 14195 14211 14230 1 14256 14269 14282
14295 14308 14321 14334 14343 14346 14352 14373 14380 1 14441 14448 14481
14484 14487 14511 14517 14523 14529 14538 14550 14562 1 14654 14664 14699
14714 14723 14747 14821 14833 14839 14866 14894 14903 1 14936 14958 14967
14973 14985 15010 15016 15022 15041

.SINO = 020000
1# 3536 3554 3616 3634 3733 3751 3813 3831 3940 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
7991 7994 14121 14131 14141 14151 14161 14171 14181 14191 14604 14623

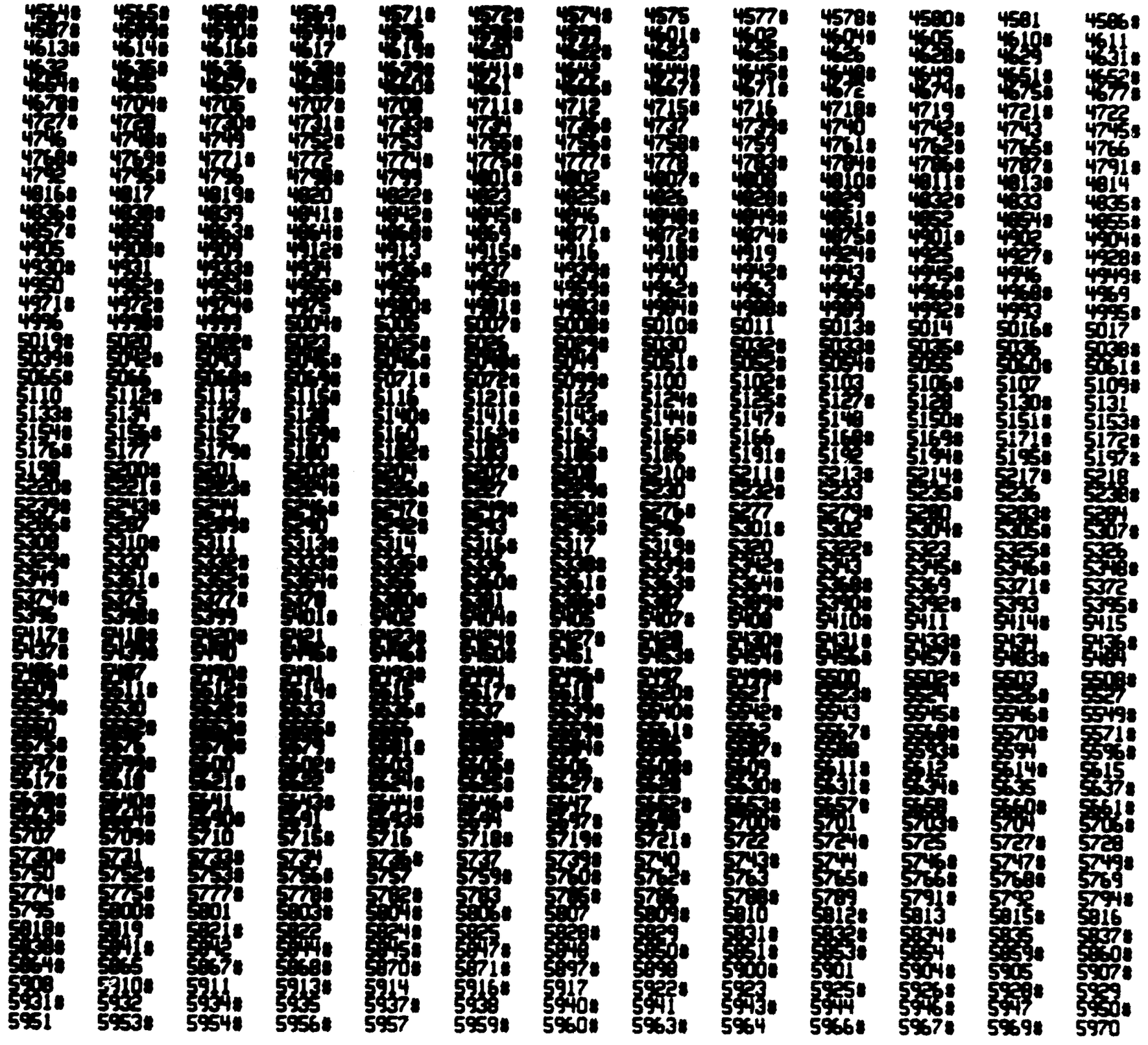
.SIN1 = 120000
1# 2297 2315 2377 2395 2494 2512 2574 2592 2691 2709 2771 2789
2888 2906 2968 2986 3098 3113 3187 3202 3315 3339 3413 3437 7407
7551 7608 7614 7641 7812 7824 7858 7985 13729 13757 13783 14049 14059
14069 14079 14089 14099 14109 14532 14544 14556 14568 14574 14580 14583 14589
14595 14598 14717 14726 14732 14738 14750 14758 14784 14824 14830 14845 14851
14900 14915 14921 14964 14979 14988 14994 15028 15034 15047

.SMEM = 040000
1# 2321 2327 2401 2407 2518 2524 2598 2604 2715 2721 2795 2801
2912 2918 2982 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 5226 5232 5336 5340 5351 5377
5420 5433 5439 5496 5542 5555 5561 5640 5646 5706 5749
5762 5768 5791 5834 5847 5853 5913 5956 5969 5975 5998 6041 6054
6060 6120 6163 6176 6182 6205 6218 6261 6267 6327 6370 6383 6389
6412 6455 6468 6474 6531 6577 6590 6596 6619 6662 6675 6681 6741
6784 6797 6803 6826 6869 6882 6888 6946 6991 7004 7010 7033 7076
7089 7095 7155 7198 7211 7217 7240 7283 7296 7302 7471 7674 7714
7815 7842 8042 8054 8060 8073 8157 8160 8167 8185 8247 8314 8317
8324 8342 8407 8477 8489 8567 8579 8727 8739 8859 8878 8907 9098
9107 9113 9160 9172 9178 9184 9415 9418 9424 9430 9477 9486 9493
9496 9501 9732 9735 9741 9747 9794 9803 9806 9812 9818 10049 10052
10058 10064 10111 10120 10123 10129 10135 10366 10369 10375 10381 10428 10437
10440 10446 10452 10683 10686 10692 10698 10745 10754 10757 10763 10769 11000

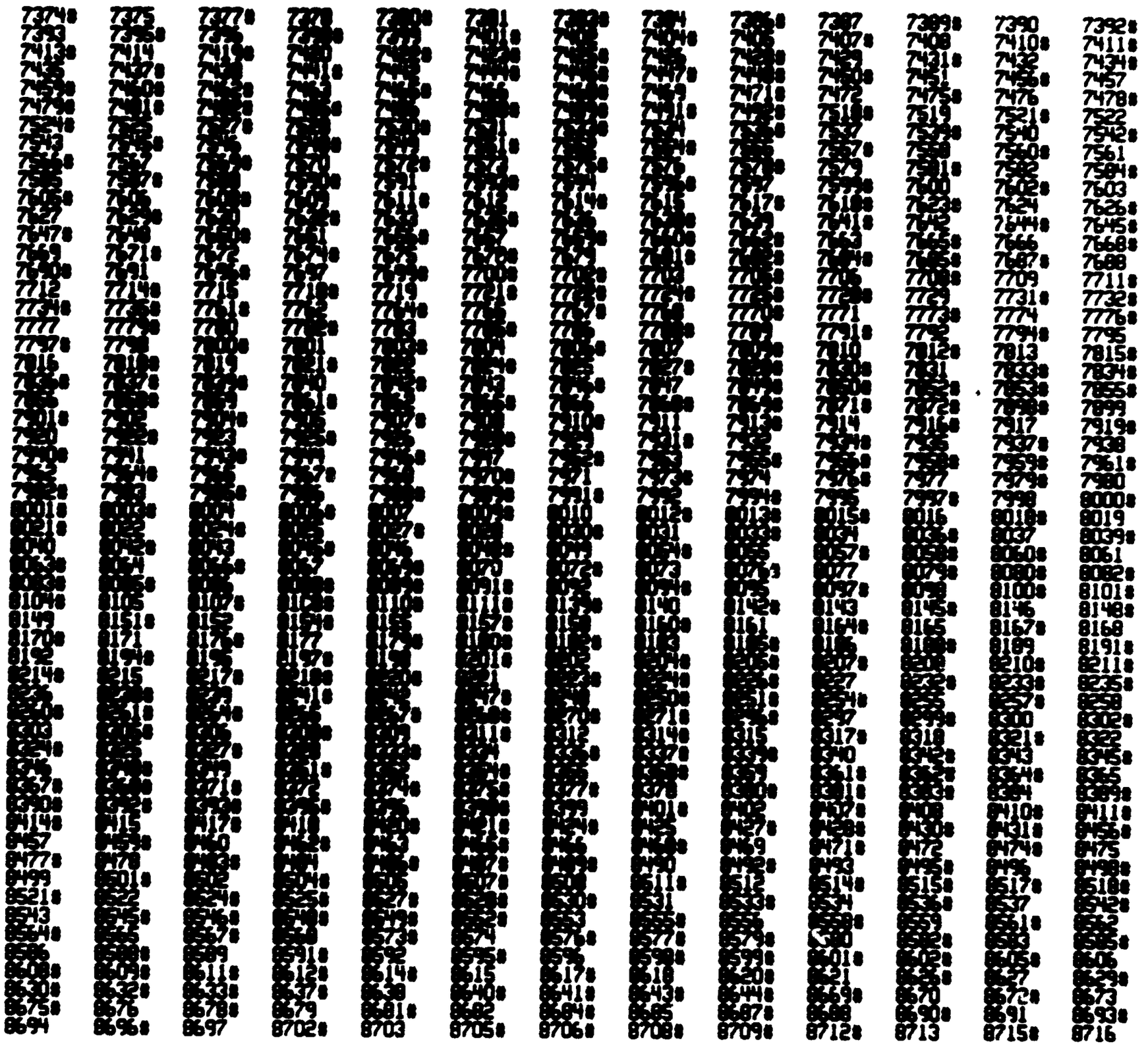
```
.START  002402
.SUBMC= 000040
.SUB1C= 000340
.SUB2C= 000360
.SO   = 020000
.TIMER  007464
.XOR  = 000320
.SASTA= ****** U
.SX   = 002034
.S.   = 104510

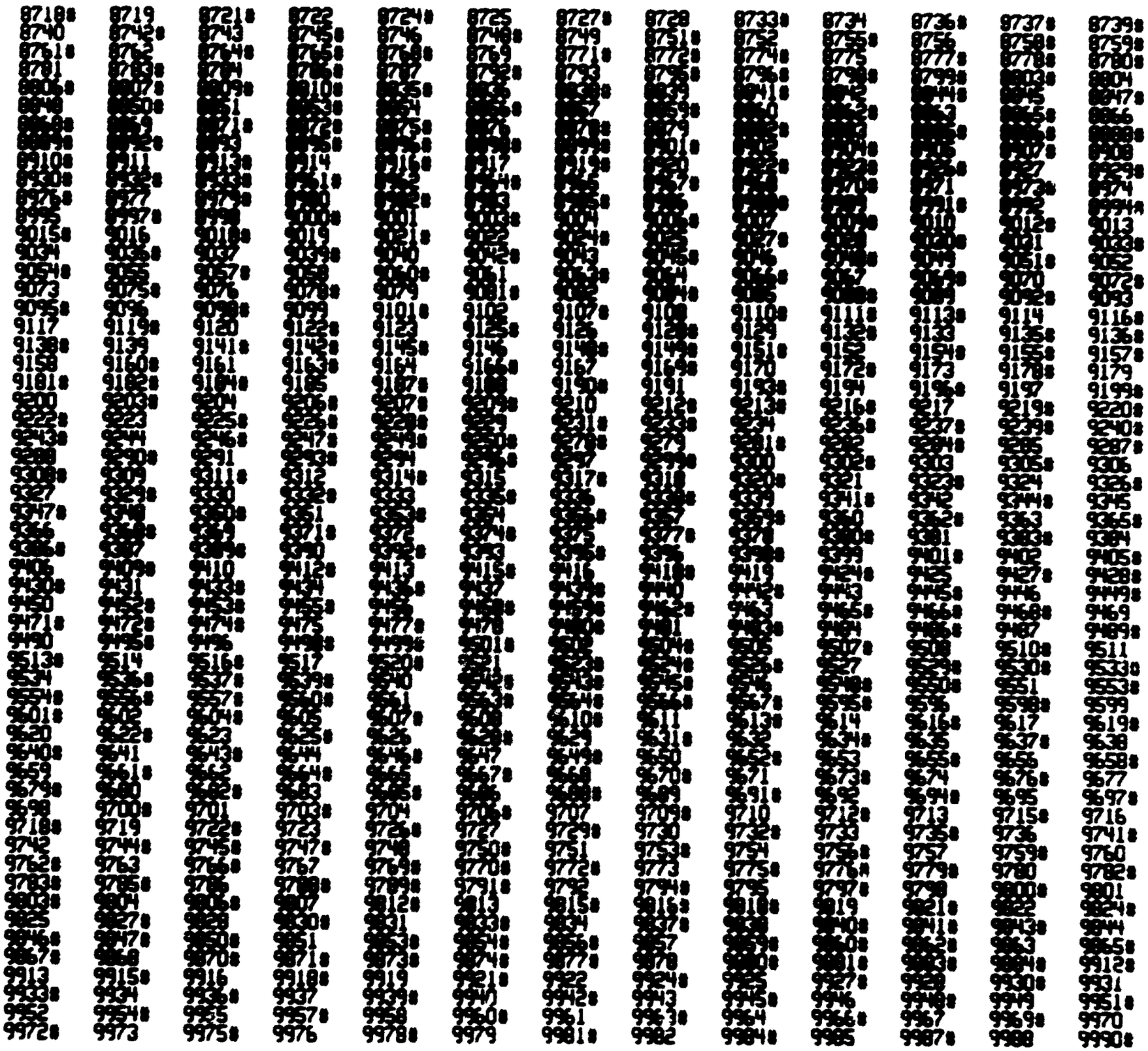
11000  11015  11062  11074  11080  11086  11317  11320  11326  11332  11379  11388  11391
11397  11403  11436  11437  11643  11649  11696  11705  11708  11714  11720  11951  11954
11960  11966  12013  12022  12025  12031  12037  12268  12277  12283  12330  12342  12348
12361  12367  12413  12420  12447  12453  12465  12671  12902  12911  12917  12964  12970
13202  13213  13219  13222  13226  13234  13281  13290  13293  13299  13305  13536  13545
13551  13557  13610  13616  13622  13738  13770  13808  14355  14367  14526
  233#  732#  748#  1828#
    1#  11362#  12002#
    1#  8542#  8656#  8702#  8792  11320  11388
    1#  2306#  2386#  2503#  2583  2700  2780  2897  2977  3104  3193  3330  3428
 3545#  3625#  3743#  3822#  3939#  4019  4136  4216  4333  4413  4530  4610  4727
 4807#  4924#  5004#  5121#  5191  5301  5386  5508  5593  5715  5800  5922  6007
 6129#  6214#  6336#  6421#  6543  6628  6750  6835  6957  7042  7164  7249  7419
 7456#  7656#  7696#  8054#  8176  8247  8333  8407  8483  8573  8733  9107  9178
 9424#  9486#  9741#  9812  10058  10129  10375  10446  10692  10763  11009  11080  11326
11397  11643  11714  11960  12031  12277  12348  12594  12665  12911  12982  13222  13228
13290  13299  13545  13616  13980  14361  14426  14613  14632
    1#
 1565#  1752#
    1#  10369#  10437
 1178#  1181#
  486#  491#
2283#  2284#  2286#  2287  2290#  2291  2294#  2295  2297#  2298  2300#  2301  2306#
2307  2309#  2310#  2312#  2313  2315#  2316  2318#  2319  2321#  2322  2324#  2325
2327#  2328#  2331#  2342#  2344#  2346#  2347#  2338  2340#  2341#  2344#  2345  2347#
2348#  2350#  2361  2363#  2364#  2366#  2357  2362#  2363#  2365#  2366#  2370#  2371
2374#  2375  2377#  2378  2380#  2381  2386#  2387  2389#  2390#  2392#  2393  2395#
2396  2398#  2399  2401#  2402  2404#  2405  2407#  2408  2411#  2412  2414#  2415#
2417#  2418  2420#  2421#  2424#  2426  2427#  2428#  2430#  2431  2433#  2434#  2436#
2437  2442#  2443#  2447#  2448  2450#  2451#  2453#  2454#  2480#  2481  2483#  2484
2487#  2488  2491#  2492  2494#  2496  2497#  2498  2503#  2504  2506#  2507#  2509#
2510  2512#  2513  2515#  2516  2519  2521#  2522  2524#  2525  2527#  2528  2529
2531#  2532#  2534#  2536#  2537#  2539#  2541#  2542#  2543#  2545#  2547#  2548  2550#
2551#  2553#  2554#  2556#  2557#  2559#  2560#  2561#  2563#  2564#  2566#  2572  2575
2577#  2578  2580#  2581#  2583#  2584#  2586#  2587#  2589#  2590#  2593  2595#  2598#
2599  2601#  2602  2604#  2605  2607#  2608#  2609  2611#  2612#  2614#  2615  2617#  2618#
2621#  2622  2624#  2625#  2627#  2628#  2630#  2631#  2633#  2634#  2636#  2639#  2644#
2645  2647#  2648#  2650#  2651#  2653#  2654#  2655#  2657#  2658#  2660#  2661  2663#
2691#  2692  2694#  2695#  2697#  2698#  2700#  2701  2703#  2704#  2706#  2707  2710  2712#
2713  2715#  2716  2718#  2719  2721#  2722  2725#  2726  2727#  2728#  2729#  2731#  2732
2734#  2736#  2737#  2739#  2741#  2742#  2744#  2745  2747#  2748#  2750#  2751  2753#  2755#
2757#  2758#  2760#  2761#  2763#  2764#  2766#  2767  2769  2771#  2772  2775  2780#  2781
2783#  2784#  2785#  2787#  2788#  2790  2792#  2793  2795#  2796#  2798#  2799  2801#  2803#
2802  2805#  2806#  2808#  2809#  2811#  2812  2814#  2815#  2817#  2818#  2819  2821#  2822#
2824#  2825#  2827#  2828#  2830#  2831  2833#  2834#  2836#  2837#  2839#  2841#  2845#  2847#
2848#  2874#  2875  2877#  2878#  2880#  2881#  2882  2884#  2885#  2887#  2888#  2891#  2892#
2897#  2898  2900#  2901#  2903#  2904  2906#  2907  2909#  2910  2912#  2913  2915#  2916
2916  2918#  2919  2921#  2923#  2925#  2926#  2927  2929#  2930#  2932#  2933  2936  2938#
2938#  2939#  2941#  2942  2944#  2945#  2947#  2948  2950#  2951#  2953#  2954  2956#  2961#
2962  2965#  2966  2968#  2969  2971#  2972  2975#  2977#  2978  2980#  2981  2983#  2984
2986#  2987  2988#  2990  2992#  2993  2995#  2996  2998#  2999  3001#  3002  3004#  3005#
3006#  3008#  3009  3011#  3012#  3015#  3016  3018#  3019#  3021#  3022  3024#  3025#
3027#  3028  3033#  3034#  3038#  3039  3041#  3042#  3044#  3045#  3072#  3073  3075#
3076  3079#  3080  3083#  3084  3086#  3087  3089#  3090  3092#  3093  3095#  3096
3098#  3099  3104#  3105  3107#  3108#  3110#  3111  3113#  3114  3116#  3117  3119#
3120  3122#  3123  3125#  3126  3129#  3130  3132#  3133#  3135#  3136  3138#  3139#
```

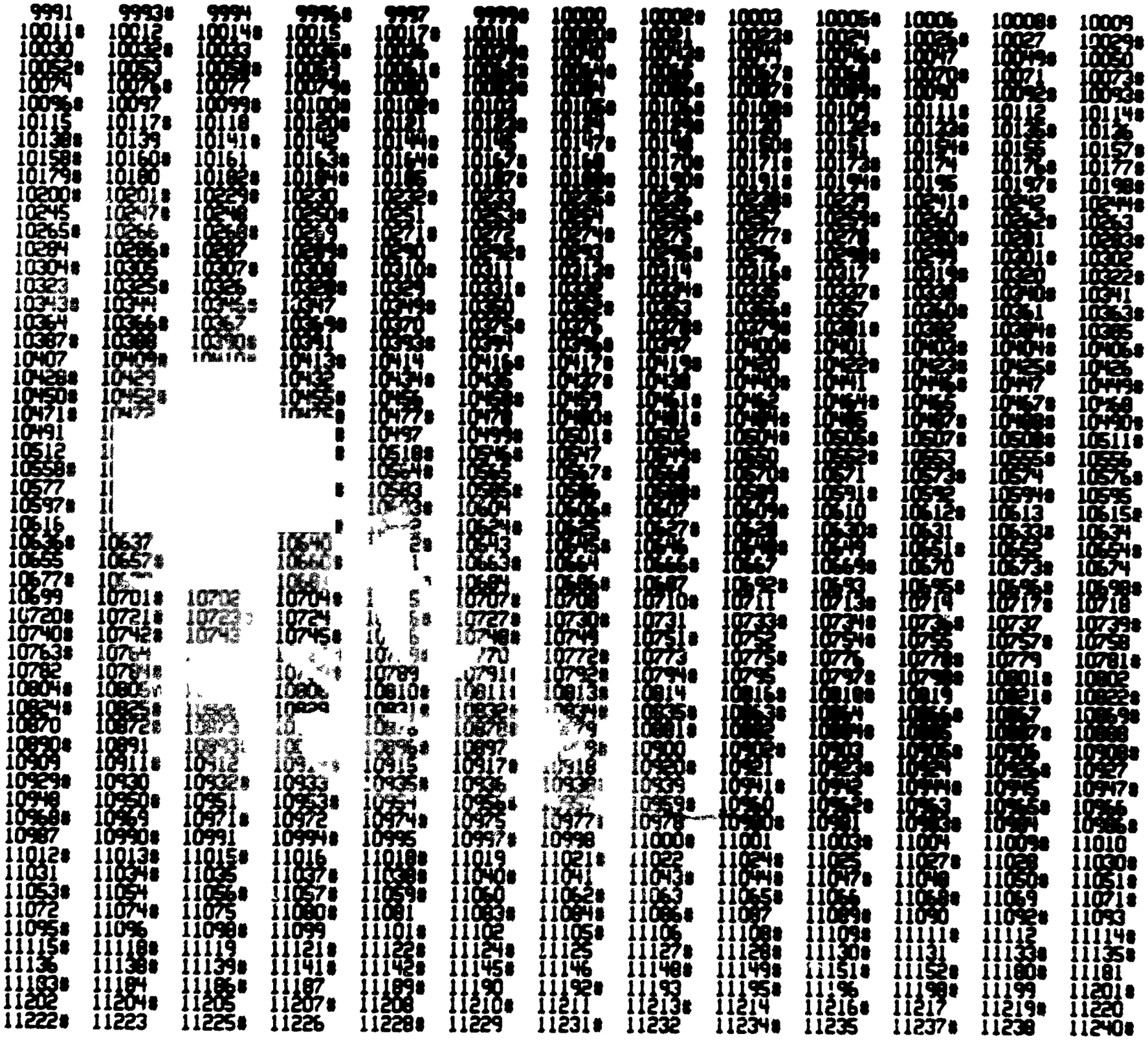
```
3142#  3143   3145#  3146#  3148#  3149   3151#  3152#  3154#  3155   3160#  3161#  3163#
3164#  3168#  3169   3172#  3173   3175#  3176   3178#  3179   3181#  3182   3184#  3185
3187#  3188   3193#  3194   3196#  3197#  3199#  3200   3202#  3203   3205#  3206   3208#
3209   3211#  3212   3214#  3215   3218#  3219   3221#  3222#  3224#  3225   3227#  3228#
3231#  3248   3248#  3255#  3237#  3238#  3240#  3241#  3243#  3244   3246#  3247#  3254#
3255   3257#  3258#  3260#  3261#  3309#  3240   3242#  3243   3246#  3247   3250#  3254#
3303#  3304   3306#  3307   3309#  3310   3312#  3313   3315#  3316   3318#  3300#  3301
3322   3324#  3325   3330#  3331   3333#  3334#  3336#  3337   3338#  3340   3342#  3343
3345#  3346   3348#  3349   3351#  3353#  3354#  3356#  3357   3358#  3361#  3362   3364#
3368#  3369   3370#  3371#  3372#  3373#  3374#  3375   3377#  3378#  3380#  3382#  3387#
3388#  3390#  3391#  3394#  3395   3398#  3399   3401#  3402   3404#  3405   3407#  3408   3410#
3411   3413#  3414   3416#  3417   3418#  3419#  3420   3422#  3423   3425#  3429   3431#  3432#
3434#  3436   3437#  3438   3440#  3441   3442#  3443#  3444   3446#  3447   3449#  3450   3453#
3454   3456#  3457#  3458#  3460   3461#  3462#  3463#  3464#  3466#  3467   3469#  3470#  3472#  3473
3523   3525#  3526#  3528#  3529#  3530#  3531#  3533#  3534#  3536#  3537   3538#  3495#  3496#  3522#
3540#  3544#  3551#  3552   3554#  3555   3557#  3558   3560#  3561   3563#  3540   3545#  3546
3567   3570#  3571   3573#  3574#  3575#  3576#  3577   3579#  3580#  3583#  3584   3586#  3587#
3589#  3590   3592#  3593#  3595#  3596#  3598#  3601#  3602#  3604#  3605#  3609#  3610   3613#
3614   3616#  3617   3619#  3620   3623#  3624   3626#  3627   3628#  3629#  3631#  3632   3634#  3635
3637#  3638   3640#  3641   3643#  3644   3646#  3647   3650#  3651   3653#  3654#  3656#
3657   3659#  3660#  3663#  3664   3666#  3667#  3669#  3670   3672#  3673#  3675#  3676
3681#  3682#  3686#  3687   3689#  3690#  3692#  3693#  3719#  3720   3722#  3723   3726#
3727   3730#  3731   3733#  3734   3736#  3737   3742#  3743   3745#  3746#  3748#  3749
3751#  3752   3754#  3756   3757#  3759   3760#  3761   3763#  3764   3767#  3768   3770#
3771#  3773#  3774   3776#  3777#  3780#  3781   3783#  3784#  3786#  3787   3789#  3790#
3792#  3793   3798#  3799#  3801#  3802#  3806#  3807   3810#  3811   3813#  3814   3816#
3817   3822#  3823   3825#  3827#  3828#  3829   3831#  3832   3834#  3835   3837#  3838
3840#  3841   3843#  3844   3847#  3848   3850#  3851#  3853#  3854   3856#  3857#  3860#
3861   3863#  3864#  3866#  3867   3869#  3870#  3872#  3873   3875#  3878#  3883#  3884
3886#  3887#  3888#  3890#  3916#  3917   3919#  3920   3923#  3924   3927#  3928   3930#
3931   3933#  3934   3936#  3940   3942#  3943#  3945#  3946   3948#  3949   3951#  3952
3954#  3956   3957#  3958#  3960#  3961#  3963#  3964   3966#  3967#  3969#  3970#  3971   3973#
3974#  3977#  3978   3980#  3981#  3983#  3984   3986#  3987#  3989#  3990   3995#  3996#
3998#  3999#  4003#  4001   4007#  4008   4009#  4010#  4011   4013#  4014   4019#  4020   4022#
4023#  4025#  4026   4028#  4029   4031#  4032   4034#  4035   4037#  4038   4040#  4041
4044#  4045   4047#  4048#  4050#  4051   4053#  4054#  4057#  4058   4060#  4061#  4063#
4064   4066#  4067#  4069#  4070   4075#  4076#  4080#  4081   4083#  4084#  4086#  4087#
4113#  4114   4116#  4117   4120#  4121   4124#  4125   4127#  4128   4130#  4131   4136#
4137   4139#  4140#  4142#  4143   4145#  4146   4148#  4149   4151#  4152   4154#  4155
4157#  4158   4161#  4162   4164#  4165#  4167#  4168   4170#  4171#  4174#  4175   4177#
4178#  4180#  4181   4183#  4184#  4186#  4187   4192#  4193#  4195#  4196#  4200#  4201
4204#  4205   4207#  4208   4210#  4211   4216#  4217   4219#  4220#  4222#  4223   4225#
4226   4228#  4229   4231#  4232   4234#  4235   4237#  4238   4241#  4243   4244#  4245#
4247#  4248   4250#  4251#  4254#  4255   4257#  4259#  4260#  4261   4263#  4264#  4266#
4267   4272#  4273#  4277#  4278   4280#  4281#  4283#  4284#  4310#  4311   4313#  4314
4317#  4318   4321#  4322   4324#  4325   4327#  4328   4333#  4334   4336#  4337#  4339#
4340   4342#  4343   4345#  4346   4348#  4349   4351#  4352   4354#  4355   4358#  4359
4361#  4362#  4364#  4365   4367#  4368#  4371#  4372   4374#  4375#  4377#  4378   4380#
4381#  4383#  4384   4389#  4390#  4392#  4393#  4397   4398   4401#  4403   4404#  4405
4407#  4408   4413#  4414   4416#  4417#  4419#  4421   4422#  4423   4425   4426   4428#
4429   4431#  4432   4434#  4435   4438#  4439   4441#  4442#  4444#  4445   4446   4448#
4451#  4452   4454#  4455#  4457#  4458   4460#  4461#  4463#  4464   4469#  4470#  4474#
4475   4477#  4478#  4480#  4481#  4507#  4508   4510#  4511   4514#  4515   4518#  4519
4521#  4522   4524#  4525   4530#  4531   4533#  4534#  4536#  4537   4539#  4540   4542#
4543   4545#  4546   4548#  4549   4551#  4552   4555#  4556   4558#  4559#  4561#  4562
```



```
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#  6056   6057#  6058#
6060#  6061   6062#  6063#  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   6235#  6236   6238#
6239   6242#  6243   6245#  6246#  6248#  6249   6251#  6252#  6256#  6260   6263#  6264#
6261#  6262   6264#  6265#  6267#  6268   6273#  6274#  6278#  6279   6281#  6282#  6284#
6286#  6311#  6312   6314#  6315   6318#  6319   6321#  6322   6324#  6326   6327#  6328
6330#  6331   6336#  6337   6339#  6340#  6342#  6343   6345#  6346   6348#  6349   6351#
6352   6354#  6355   6357#  6358   6360#  6361   6364#  6365   6367#  6368   6370#  6371
6373#  6374#  6377#  6378   6380#  6381#  6383#  6384   6386#  6387#  6388#  6390   6395#
6396#  6398#  6399#  6403#  6404   6406#  6407   6409#  6410   6412#  6413   6415#  6416
6421#  6422   6424#  6425#  6427#  6428   6430#  6431   6433#  6434   6436#  6437   6439#
6440   6442#  6443   6445#  6446   6449#  6450   6452#  6453#  6455#  6456   6458#  6459#
6462#  6463   6465#  6466#  6468#  6469   6471#  6472#  6474#  6475   6480#  6481#  6485#
6486   6488#  6489#  6491#  6492#  6518#  6519   6521#  6522   6525#  6526   6528#  6529
6531#  6532   6534#  6536   6537#  6538   6543#  6544   6546#  6547#  6550   6551#  6552#
6553   6555#  6556   6558#  6559   6561#  6562   6564#  6565   6567#  6568   6571#  6572
6574#  6575#  6577#  6578   6580#  6581#  6584#  6585   6587#  6588#  6590   6591   6593#
6594#  6596#  6597   6602#  6603#  6605#  6606#  6610#  6611   6613#  6614   6616#  6617
6619#  6620   6622#  6623   6628#  6629   6631#  6634#  6635   6637#  6638   6639#  6640#
6641   6643#  6644   6646#  6647   6649#  6650   6652#  6653   6655#  6657   6658#  6660#
6662#  6663   6665#  6666#  6669#  6670   6672#  6673#  6675#  6676   6678#  6679#  6681#
6682   6687#  6688#  6690#  6693#  6695#  6696#  6698#  6699#  6725#  6726   6728#  6729
6732#  6733   6736#  6737#  6738#  6739   6741#  6743   6744#  6745   6750#  6751   6753#
6754#  6756#  6757   6759   6760   6762#  6763   6765   6766   6768#  6769   6771#  6772
6774#  6775   6778#  6779   6781#  6782#  6784#  6785   6787#  6788#  6791#  6792   6794#
6795#  6797#  6798#  6800#  6801#  6803#  6804   6806#  6810#  6812#  6813#  6815#  6818
6820#  6821   6823#  6824   6827#  6828#  6830   6831   6833#  6834   6836#  6837#  6841#
6842   6844#  6845   6847#  6848#  6850#  6851   6853#  6854   6856#  6857   6858#  6860
6863#  6864   6866#  6867#  6869#  6870   6872#  6873#  6876#  6877   6879#  6880#  6882#
6883   6885#  6886#  6888#  6889#  6891#  6892#  6893#  6900   6902#  6903#  6905#  6906#
6932#  6933   6935#  6936#  6938#  6940   6942#  6943   6945#  6946   6948#  6949   6951#
6952   6957#  6958   6960#  6961#  6963#  6964   6966#  6967   6969#  6970   6972#  6973
6975#  6976   6978#  6979   6981#  6982   6985#  6986   6988#  6989#  6991#  6992   6994#
6995#  6998#  6999   7001#  7002#  7004#  7005   7007#  7008#  7010#  7011   7016#  7017#
7019#  7020#  7024#  7025   7027#  7028   7030#  7031   7033#  7034   7036#  7037   7042#
7043   7045#  7046#  7048#  7049   7051#  7052   7054#  7055   7057#  7058   7060#  7061
7063#  7064   7066#  7067   7070#  7071   7073#  7074#  7076#  7077   7079#  7080#  7083#
7084   7086#  7087#  7089#  7090   7092#  7093#  7095#  7096   7101#  7102#  7106#  7107
7109#  7110#  7112#  7113#  7139#  7140   7142#  7143   7146#  7147   7149#  7150   7152#
7153   7155#  7156   7158#  7159   7164#  7165   7167#  7168#  7170#  7171   7173#  7174
7176#  7177   7179#  7180   7182#  7183   7185#  7186   7188#  7189   7192#  7193   7195#
7196#  7198#  7199   7201#  7202#  7205#  7206   7208#  7209#  7211#  7212   7214#  7215#
7217#  7218   7223#  7224#  7226#  7227#  7231#  7232   7234#  7235   7237#  7238   7240#
7241   7243#  7244   7249#  7250   7252#  7253#  7255#  7256   7258#  7259   7261#  7262
7264#  7265   7267#  7268   7270#  7271   7273#  7274   7277#  7278   7280#  7281#  7283#
7284   7286#  7287#  7290#  7291   7293#  7294#  7296#  7297   7299#  7300#  7302#  7303
7308#  7309#  7313#  7314   7316#  7317#  7319#  7320#  7344#  7345   7347#  7348   7350#
7351   7353#  7354   7356#  7357   7359#  7360   7362#  7363   7365#  7366   7368#  7369
```

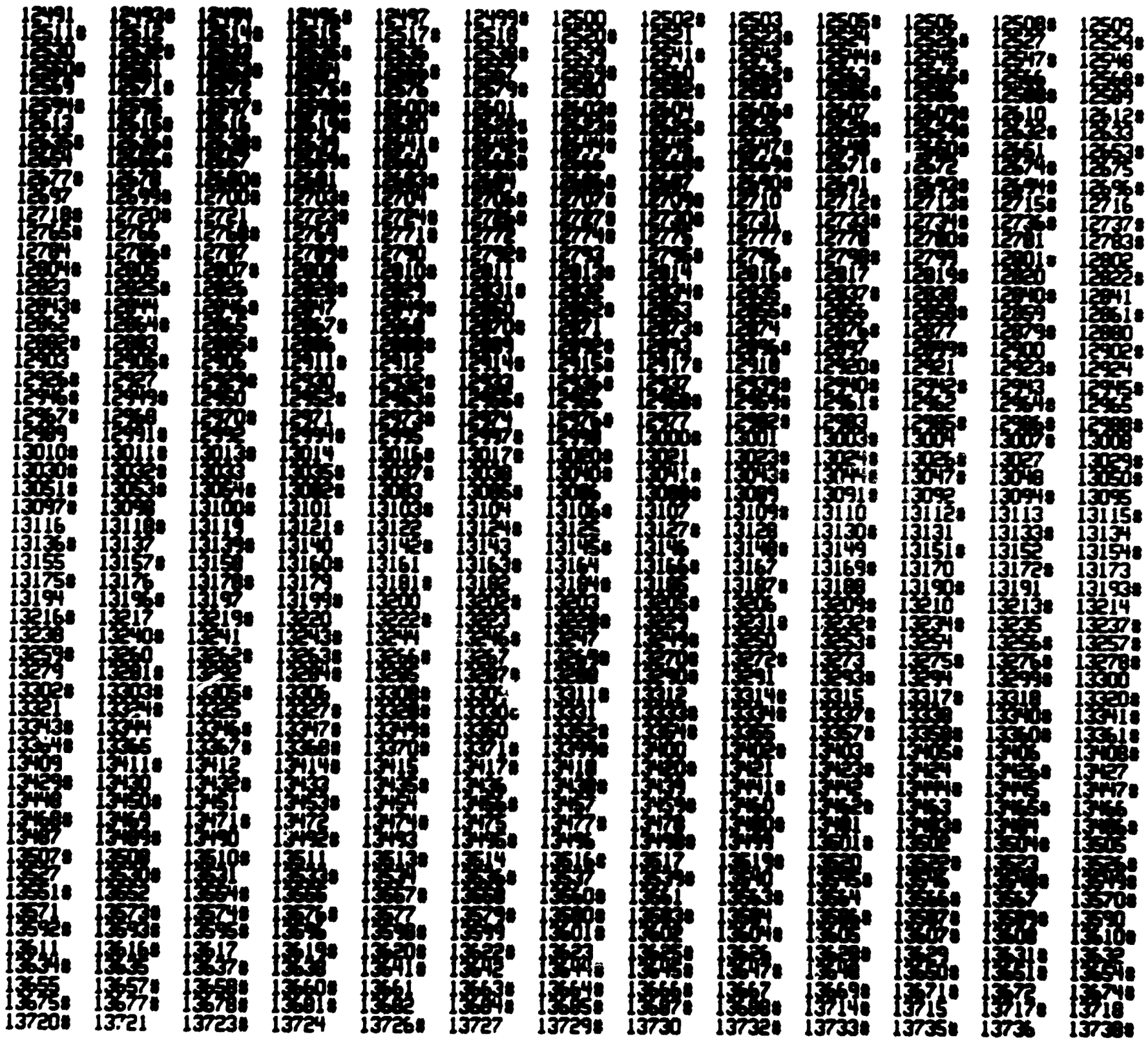






```
11241   11243#  11244   11246#  11247   11249#  11250   11252#  11253   11255#  11256   11258#  11259
11261#  11262   11264#  11265   11267#  11268   11270#  11271   11273#  11274   11276#  11277   11279#
11280   11282#  11283   11285   11286   11288   11289   11291#  11292   11294#  11295   11297#  11298
11300#  11301   11303#  11304   11307#  11308   11311#  11312   11314#  11315   11317#  11318   11320#
11321   11325#  11327   11329   11330#  11332#  11333   11335#  11336   11338#  11339   11341#  11342
11344#  11345   11347#  11348   11351#  11352   11354#  11355#  11357#  11358   11360#  11361#  11364#
11365   11367#  11368#  11370#  11371   11373#  11374#  11376#  11377   11379#  11380   11382#  11383
11385#  11386   11388   11389   11391#  11392   11397#  11398   11400#  11401#  11403#  11404   11406#
11407   11409#  11410   11412#  11413   11415#  11416   11418#  11419   11422#  11423   11425#  11426#
11428#  11429   11431#  11433#  11436#  11437   11438#  11439#  11441#  11442   11444#  11445#  11447#
11448   11450#  11452#  11453   11456#  11457   11458#  11459#  11462#  11463   11465#  11466#  11468#
11469#  11497#  11498   11500#  11501   11503#  11504   11506#  11507   11509#  11510   11512#  11513
11515#  11516   11518#  11519   11521#  11522   11524#  11526#  11527#  11528   11530#  11531   11533#
11534   11536#  11537   11539#  11540   11542#  11543   11545#  11546   11548#  11549   11551#  11552
11554#  11555   11557#  11558   11560#  11561   11563#  11564   11566#  11567   11569#  11570   11572#
11573   11575#  11576   11578#  11579   11581#  11582   11584#  11585   11587#  11588   11590#  11591
11593#  11594   11596#  11597   11599#  11600   11602#  11603   11605#  11606   11608#  11609   11611#
11612   11614#  11615   11617#  11618   11620#  11621   11624#  11625   11628#  11629   11631#  11632
11634#  11635   11637#  11638   11643#  11644   11646#  11647#  11649#  11650   11652#  11653   11655#
11656   11658#  11659   11661#  11662   11664#  11665   11668#  11669   11671#  11672#  11674#  11675
11677#  11678#  11681#  11682   11684#  11685#  11687#  11688   11690#  11691#  11693#  11694   11696#
11697   11699#  11700   11702#  11703   11705#  11706   11708#  11709   11714#  11715   11717#  11718#
11720#  11721   11723#  11724   11726#  11727   11729#  11730   11732#  11733   11735#  11736   11739#
11740   11742#  11743#  11745#  11746   11748#  11749#  11752#  11753   11755#  11756#  11758#  11759
11761#  11762#  11764#  11765   11767#  11769#  11770   11772#  11773#  11775#  11776#  11779#  11780
11782#  11783#  11785#  11786#  11814#  11815   11817#  11818   11820#  11821   11823#  11824   11826#
11827   11829#  11830   11832#  11833   11835#  11836   11838#  11839   11841#  11842   11844#  11845
11847#  11848   11850#  11851   11853#  11854   11856   11857   11859#  11860   11862#  11863   11865#
11866   11868#  11869   11871#  11872   11874#  11875   11877#  11878   11880#  11881   11883#  11884
11886#  11887   11889#  11890   11892#  11893   11895#  11896   11898#  11899   11901#  11902   11904#
11905   11907#  11908   11910#  11911   11913#  11914   11916#  11917   11919#  11920   11922#  11923
11925#  11926   11928#  11929   11931   11932   11934#  11936   11937#  11938   11941#  11942   11945#
11946   11948#  11949   11951#  11952   11954#  11955   11960#  11961   11963#  11964#  11966#  11967
11969#  11970   11972#  11973   11975   11976   11978#  11979   11981#  11982   11985#  11986   11988#
11989#  11991#  11992   11994#  11996   11998   11999   12001#  12002#  12004#  12005   12007#  12008#
12010#  12011   12013#  12014   12016   12017   12019#  12020   12022#  12023   12025#  12026   12031#
12032   12034#  12035#  12037#  12038   12040#  12041   12043#  12044   12046#  12047   12049#  12050
12052#  12053   12055#  12057   12059#  12060#  12062#  12063   12065#  12066#  12067   12069#  12070   12072#
12073#  12075#  12076   12078#  12079#  12081#  12082   12084#  12085#  12087   12089#  12090#  12092#
12093#  12096#  12097   12099#  12100#  12102#  12103   12131#  12132   12134#  12135   12137#  12138
12140#  12141   12143#  12144   12146#  12147   12150   12152#  12153#  12155#  12156   12158#
12159   12161#  12162   12164#  12165   12167#  12168   12170#  12171   12173#  12174   12176#  12177
12179#  12180   12182#  12183   12185#  12186   12188   12189   12191#  12192   12194#  12195   12197#
12198   12200#  12201   12203#  12204   12206#  12207   12209#  12210   12212#  12213   12215#  12216
12218#  12219   12221#  12222   12224#  12225   12227#  12228   12230#  12231   12233#  12234   12236#
12237   12239#  12240   12242#  12243   12245#  12246   12248#  12249   12251#  12252   12254#  12255
12258#  12259   12262#  12263   12265#  12266   12268#  12269   12271#  12272   12275#  12276   12278   12280#
12281#  12283#  12284   12286#  12287   12289#  12290   12292#  12293   12295#  12296   12298#  12299
12302#  12303   12305#  12306#  12308#  12309   12311#  12312#  12315#  12316   12318#  12319#  12321#
12322   12324#  12325#  12327#  12328   12330#  12331   12333#  12334   12336#  12337   12339#  12340
12342#  12343   12346#  12349   12351#  12352#  12354#  12355   12357#  12358   12360#  12361   12363#
12364   12366#  12367   12369#  12370   12372#  12374   12376#  12377#  12379#  12380   12382#  12383#
12386#  12387   12389#  12390#  12392#  12393   12395#  12396#  12398#  12399   12401#  12403#  12404
12406#  12407#  12409#  12410#  12413#  12414   12416#  12417#  12419#  12420#  12448#  12449   12451#
12452   12454#  12455   12457#  12458   12460#  12461   12463#  12464   12466#  12467   12469#  12470
12472#  12473   12475#  12476   12478#  12479   12481#  12482   12484#  12485   12487#  12488   12490#
```

F12

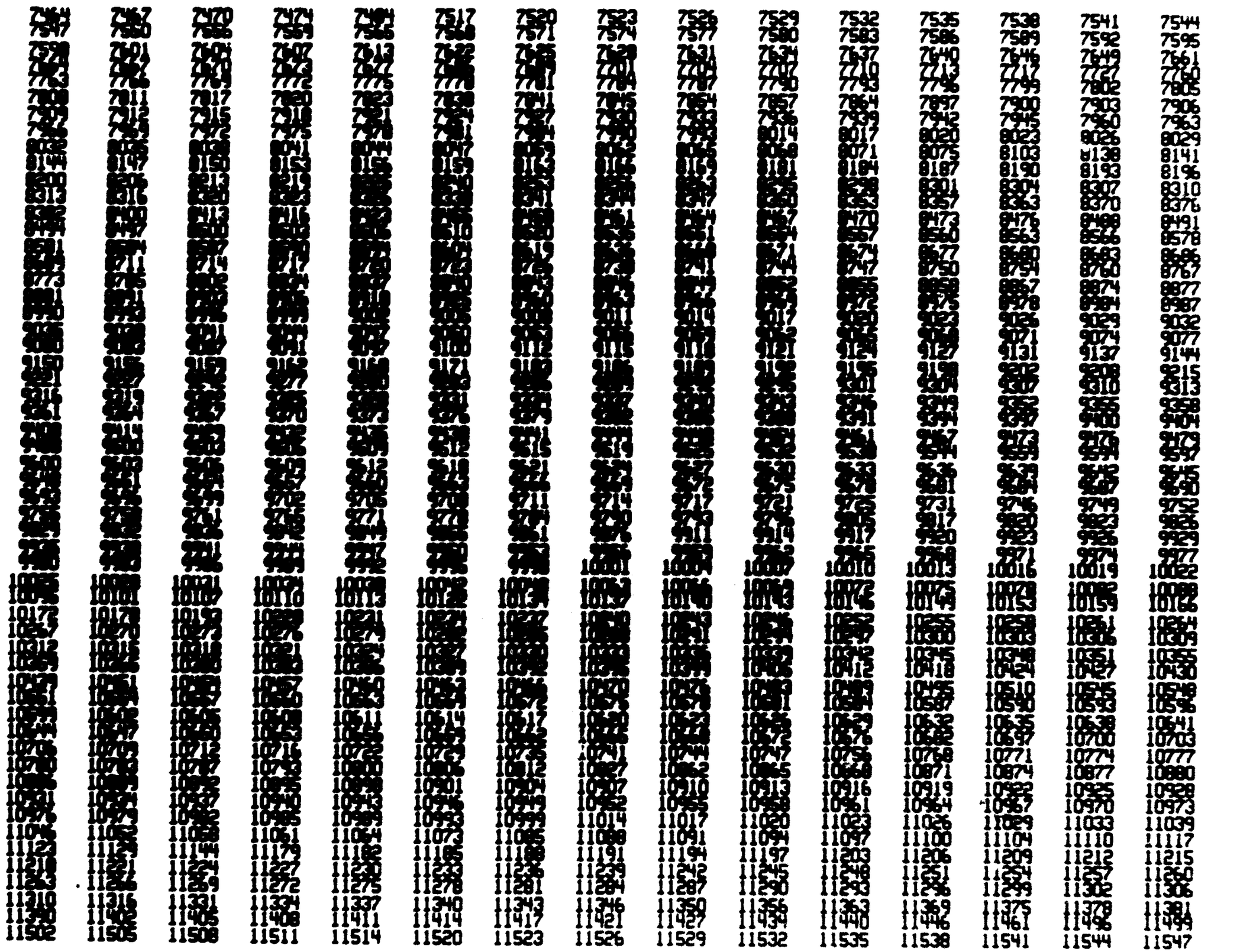


13739 13742# 13743 13745# 13746# 13748# 13749# 13751# 13752 13754# 13755 13757# 13758
13760# 13761# 13763# 13764# 13767# 13768 13770# 13771 13774# 13775 13777# 13778# 13780#
13781# 13783# 13784 13786# 13787# 13789# 13790 13792# 13793 13795# 13796 13798# 13799#
13801# 13802# 13805# 13806 13808# 13809 13812# 13813 13815# 13816# 13818# 13819# 13822#
13823 13825# 13826# 13828# 13829# 13867# 13868# 13870# 13871 13873# 13874 13877 13879# 13880 13882# 13883 13885# 13886 13867
13889 13892# 13893 13895# 13896 13900# 13901 13903# 13904 13906# 13907 13909# 13910
13912# 13913 13915# 13916 13918# 13919 13921# 13922 13924# 13927# 13928 13930# 13931 13933#
13934 13936# 13937 13939# 13940 13942# 13943 13945# 13946 13948# 13949 13951# 13952
13954# 13955 13957# 13958 13960# 13963 13965# 13966 13968# 13971# 13972 13974#
13975 13980# 13981 13983# 13984# 13986# 13987# 13990# 13991 13993# 13994# 13996# 13997
13999# 14000# 14003# 14004 14006# 14007# 14009# 14010 14012# 14013 14015# 14016# 14018#
14019# 14021# 14022 14024# 14025# 14027# 14028 14031# 14032 14034# 14035 14037# 14038
14040# 14041# 14046# 14047 14049# 14050 14053# 14054 14056# 14057# 14059# 14060 14063#
14064 14066# 14067# 14069# 14070 14073# 14074 14076# 14077# 14079# 14080 14083# 14084
14086# 14087# 14089# 14090 14093# 14094 14096# 14097# 14099# 14100 14103# 14104 14106#
14107# 14109# 14110 14113# 14114 14116# 14117# 14121# 14122 14125# 14126 14128# 14129#
14131# 14132 14135# 14137 14138# 14139# 14141# 14142 14145# 14146 14148# 14149# 14151#
14152 14155# 14156 14158# 14159# 14161# 14162 14165# 14166 14168# 14169# 14171# 14172
14175# 14177 14178# 14179# 14181# 14182 14185# 14186 14188# 14189# 14191# 14192 14195#
14196 14198# 14199# 14200# 14201# 14205 14207# 14208 14211# 14212 14214# 14215# 14217# 14218
14220# 14221 14223# 14224 14226# 14227 14230# 14231 14233# 14234# 14236# 14237 14239#
14240 14243# 14244 14246# 14247# 14249# 14250 14252# 14253 14255# 14257 14259# 14260#
14262# 14263 14265# 14266 14268# 14270 14272# 14273# 14275# 14276 14278# 14279 14282#
14283 14285# 14286# 14288# 14289 14291# 14292 14294# 14295 14298# 14299# 14301# 14302
14304# 14305 14308# 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 14355# 14356 14361# 14362 14364# 14365# 14367# 14368 14370#
14371 14373# 14374 14376# 14377 14380# 14381 14383# 14384# 14386# 14387# 14389# 14390
14392# 14393# 14395# 14396# 14398# 14399 14401# 14402# 14404# 14405# 14408# 14409 14411#
14412# 14414# 14415# 14417# 14418 14420# 14421 14425# 14427 14429# 14430# 14432# 14433
14435# 14436 14438# 14439 14441# 14442 14444# 14445 14447# 14448 14450# 14451# 14452# 14455#
14456# 14457# 14459# 14460 14462# 14463 14465# 14466 14468# 14469 14471# 14472# 14474 14475#
14477# 14478# 14480# 14481 14483# 14484 14486# 14487# 14489# 14490 14492# 14493 14494 14496# 14497
14499# 14500# 14502# 14503 14505# 14506 14508# 14509 14511# 14512 14514# 14515 14517#
14518 14520# 14521 14523# 14524 14526# 14527 14530# 14531 14533# 14534 14536#
14537# 14539# 14540 14542# 14543 14545# 14546 14548# 14549 14551# 14552# 14554 14556#
14557 14560# 14561# 14563# 14564 14566# 14567 14569# 14570 14572# 14574# 14575
14577# 14578# 14580# 14581 14583# 14584 14586# 14587# 14590 14592# 14593# 14595#
14596 14598# 14599# 14601# 14602# 14604# 14605 14607# 14608 14610# 14611# 14613# 14614 14616# 14617#
14619# 14620# 14622# 14623 14625# 14626 14628# 14629 14631# 14632# 14634# 14635 14637# 14638 14639 14641#
14642 14644# 14645 14647# 14648 14650# 14651 14653# 14654 14656# 14657# 14658# 14660# 14661#
14663# 14664# 14666 14667# 14668 14670# 14671# 14673# 14674 14676# 14677 14679# 14680 14682# 14683#
14709 14711# 14712 14714# 14715 14717# 14718 14720# 14721 14723# 14724 14726# 14727
14729# 14730 14732# 14733 14735# 14736# 14738# 14739 14741# 14742# 14747# 14748 14750#
14751 14753# 14754 14756# 14757 14761# 14762# 14764# 14765# 14767# 14768 14770# 14771
14773# 14774 14776# 14777# 14779# 14780# 14784# 14785 14787# 14788# 14809# 14810 14812#
14813 14815# 14816 14818# 14819 14821# 14822 14824# 14825 14827# 14828 14830# 14831
14833# 14834 14836# 14837 14839# 14840 14842# 14843 14845# 14846 14848# 14849# 14851#
14852 14854# 14855# 14857# 14858# 14860# 14861 14863# 14864 14866# 14867 14869# 14870
14872# 14873 14875# 14876# 14878# 14879 14881# 14882 14884# 14885 14887# 14888 14890# 14891# 14892 14894# 14895 14897#
14898 14900# 14901 14903# 14904 14906# 14907 14909# 14910 14912# 14913 14915# 14916
14918# 14919# 14921# 14922 14924# 14925# 14927# 14928# 14930# 14931 14933# 14934 14936#
14937 14939# 14940 14942# 14943 14945# 14946# 14948# 14949 14951# 14952# 14954 14955#
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#

```
                    15029  15031#  15032#  15034#  15035   15037#  15038#  15041#  15042   15044#  15045#  15047#  15048
                    15050#  15051#
.2A    = 000120        1#  12588   12656
.2AWC  = 000140        1#   5159    5229   11003   11071
```

```
ADDMC      1#  11636  11704
ADD1       1#
AND        1#   3091   3180   3308   3406   7860   9734   9802  14534  14835  14905  14969  14990
BBO        1#   7409   7616   7643   7826   7987   8870   8921  14570  14853  15036
BBI        1#  14576  14591  14923
BBE        1#  13731  13759  13786  14558  14740
BBT        1#   2361   2441   2558   2638   2755   2835   2952   3032   3159   3248   3385   3483   3600   3680
           3797   3877   3994   4074   4191   4271   4388   4468   4585   4665   4782   4862   4979   5059   5167
           5237   5359   5444   5566   5651   5773   5858   5980   6065   6187   6272   6394   6479   6601   6686
           6808   6893   7015   7100   7222   7307   8231   8388  13797  14546  14734  14760  14847  14856  14917
           14926  14981  14996  15030
BC         1#  7653   8249   8409   8544   8628   8704   8794  13982  14014  14391  14775
BZ         1#  2488   2586   2505   2585   2702   2782   2899   2979   3106   3195   3332   3430   3547   3627
           3744   3824   3941   4021   4138   4218   4335   4415   4532   4612   4729   4809   4926   5006   5123
           5193   5303   5388   5510   5595   5717   5802   5924   6009   6131   6216   6338   6423   6545   6630
           6752   6852   6959   7079   7166   7251   7421   7458   7658   7698   7864   7999   8085   8087   8178
           8296   8405   8575   8736   9109   9180   9235   9426   9447   9552   9743   9814   9869  10060  10131
          10196  10377  10446  10503  10694  10765  10820  11011  11082  11137  11328  11389  11454  11645  11716
          11771  11962  12033  12088  12279  12350  12405  12596  12667  12722  12913  12984  13039  13230  13301
          13356  13547  13618  13673  14005  14363  14400  14428  14615  14634
CALL       1#  2403   2492   2549   2625   2695   2536   2539   2706   2819   2642   2723   2736   2803   2816
           2883   2963   3033   3090   3013   3036   3127   3140   3216   3229   3263   3336   3366   3451   3464
           3581   3661   3684   3765   3778   3845   3858   3881   3962   3975   4042   4055
           4159   4172   4239   4252   4275   4356   4369   4436   4449   4472   4553   4566   4633   4646
           4750   4763   4830   4843   4866   4947   4960   5027   5040   5063   5136   5145   5205   5215
           5327   5340   5412   5425   5448   5534   5547   5619   5632   5655   5741   5754   5826   5839
           5961   6033   6046   6069   6155   6168   6240   6253   6276   6362   6375   6447   6460
           6582   6654   6667   6690   6776   6789   6861   6874   6897   6983   6996   7068   7081
           7203   7275   7288   7311   7438   7473   7483   7676   7716   7726   7844   7863   8074
           8212   8262   8356   8369   8422   8504   8519   8593   8603   8635   8753   8766   8801
           8924   9130   9143   9201   9214   9241   9447   9460   9518   9531   9558   9764   9777
           9875  10081  10094  10152  10165  10192  10398  10411  10469  10482  10589  10715  10728
          10799  10826  11032  11045  11103  11116  11143  11349  11362  11420  11433  11460  11666  11679
          11750  11777  11983  11996  12054  12067  12094  12300  12313  12371  12384  12411  12617  12630
          12701  12728  12934  12947  13005  13018  13045  13251  13264  13322  13335  13362  13568  13581
          13652  13679  13740  13772  13810  13820  14051  14061  14071  14081  14091  14101  14111  14123
          14143  14153  14163  14173  14183  14193  14209  14228  14241  14254  14267  14280  14293  14306
          14332  14378  14406  14446  14652  14662  15039
COMMEN    176#
DISP       1#
ENDCOM    176#
ERROR      70#  2277   2474   2671   2868   3066   3283   3516   3713   3910   4107   4304   4501   4698   4895
           5093   5270   5477   5684   5891   6098   6305   6512   6719   6926   7133   7339   7513   7756   7893
           8133   8290   8451   8664   8830   8956   9273   9590   9907  10224  10541  10858  11175  11492  11809
          12126  12443  12760  13077  13394  13709  13852  14476  14694  15095  15119  15135  15142
ESCAPE    176#
GETPRI    176#
GETSWR    176#
HLT       189#
KMEND      1#   953
KMFRNT     1#    30
LORN       1#  9417   9485
MOVE       1#  2282   2285   2289   2293   2296   2299   2311   2314   2317   2320   2323   2326   2330   2336
           2343   2349   2355   2369   2373   2376   2379   2391   2394   2397   2400   2403   2406   2410   2416
           2423   2429   2435   2446   2479   2482   2486   2490   2493   2496   2508   2511   2514   2517   2520
           2523   2527   2533   2540   2546   2552   2566   2570   2573   2576   2588   2591   2594   2597   2600
```

2603	2607	2613	2620	2626	2632	2643	2676	2679	2683	2687	2690	2693	2705	2708
2711	2714	2717	2720	2724	2730	2737	2743	2749	2763	2767	2770	2773	2785	2788
2791	2794	2797	2800	2804	2810	2817	2823	2860	2870	2873	2876	2880	2884	2887
2890	2893	2905	2908	2911	2914	2917	2921	2957	2944	2948	2946	2960	2964	2967
2970	2983	2985	2988	2991	2994	2997	3001	3007	3014	3020	3026	3037	3071	3074
3078	3082	3085	3088	3094	3097	3109	3112	3115	3118	3121	3124	3128	3134	3141
3147	3153	3167	3171	3174	3177	3183	3186	3198	3201	3202	3207	3210	3213	3217
3223	3230	3236	3242	3253	3288	3291	3295	3299	3302	3305	3311	3314	3317	3320
3336	3338	3341	3344	3347	3350	3361	3365	3372	3373	3379	3383	3397	3400	3403
3409	3412	3415	3418	3433	3436	3439	3442	3445	3448	3452	3458	3465	3471	3477
3488	3521	3524	3528	3532	3535	3538	3550	3553	3556	3559	3566	3565	3569	3575
3580	3588	3594	3606	3612	3615	3618	3630	3633	3638	3643	3652	3645	3649	3655
3663	3668	3674	3686	3718	3721	3725	3728	3732	3735	3747	3750	3753	3756	3759
3763	3766	3772	3779	3785	3791	3805	3809	3812	3815	3827	3830	3833	3836	3839
3842	3846	3852	3859	3865	3871	3882	3915	3918	3922	3926	3929	3932	3944	3947
3950	3963	3966	3969	3973	3976	3979	3982	3988	4003	4006	4009	4012	4021	4027
4030	4033	4036	4039	4043	4049	4056	4062	4068	4079	4112	4115	4119	4123	4126
4129	4141	4144	4147	4150	4153	4156	4160	4166	4173	4179	4185	4199	4203	4206
4209	4221	4224	4227	4230	4233	4236	4240	4246	4253	4259	4265	4276	4309	4312
4316	4320	4323	4326	4338	4341	4344	4347	4350	4353	4357	4363	4370	4376	4382
4396	4400	4403	4406	4418	4421	4424	4427	4430	4433	4437	4443	4450	4456	4462
4473	4506	4509	4513	4517	4520	4523	4536	4538	4541	4544	4547	4550	4554	4560
4567	4573	4579	4593	4597	4600	4603	4615	4618	4621	4624	4627	4630	4634	4640
4647	4653	4659	4670	4703	4706	4710	4714	4717	4720	4732	4735	4738	4741	4744
4747	4751	4757	4764	4770	4776	4790	4794	4797	4800	4812	4815	4818	4821	4824
4827	4831	4837	4844	4850	4856	4867	4900	4903	4907	4911	4914	4917	4929	4932
4935	4938	4941	4944	4948	4954	4961	4967	4973	4987	4991	4994	4997	5009	5012
5015	5018	5021	5024	5028	5034	5041	5047	5053	5064	5098	5101	5105	5108	5111
5114	5126	5129	5132	5136	5146	5155	5164	5175	5178	5181	5184	5196	5199	5202
5208	5216	5225	5234	5242	5275	5278	5282	5285	5288	5291	5294	5306	5309	5312
5315	5318	5321	5324	5328	5334	5341	5347	5353	5367	5370	5373	5376	5379	5391
5394	5397	5400	5403	5406	5409	5413	5419	5426	5432	5438	5446	5482	5485	5489
5492	5495	5498	5501	5513	5516	5519	5522	5526	5528	5531	5535	5541	5546	5554
5560	5574	5577	5580	5583	5586	5598	5601	5604	5607	5610	5613	5616	5620	5626
5633	5639	5645	5656	5689	5692	5696	5699	5702	5705	5708	5720	5723	5726	5729
5732	5735	5738	5742	5746	5755	5761	5767	5781	5784	5787	5790	5793	5805	5808
5811	5814	5817	5820	5823	5827	5833	5840	5846	5852	5863	5896	5899	5903	5906
5909	5912	5915	5927	5930	5933	5936	5939	5942	5945	5949	5955	5962	5965	5974
5988	5991	5994	5997	6000	6012	6015	6018	6021	6024	6027	6030	6034	6040	6047
6053	6059	6070	6103	6106	6110	6113	6116	6119	6122	6134	6137	6140	6143	6146
6149	6152	6156	6162	6169	6175	6181	6195	6198	6201	6204	6207	6219	6222	6225
6228	6231	6234	6237	6241	6247	6254	6260	6266	6277	6310	6313	6317	6320	6323
6326	6329	6341	6344	6347	6350	6353	6356	6359	6363	6369	6376	6382	6388	6402
6405	6408	6411	6414	6426	6429	6432	6435	6438	6441	6444	6448	6454	6461	6467
6473	6484	6517	6520	6524	6527	6530	6533	6536	6548	6551	6554	6557	6560	6563
6566	6570	6576	6583	6589	6595	6609	6612	6615	6618	6621	6633	6636	6639	6642
6645	6648	6651	6655	6661	6668	6674	6680	6691	6724	6727	6731	6734	6737	6740
6743	6755	6758	6761	6764	6767	6770	6773	6777	6783	6790	6796	6802	6816	6819
6822	6825	6828	6840	6843	6846	6849	6852	6855	6858	6862	6868	6875	6881	6887
6898	6931	6934	6938	6941	6944	6947	6950	6962	6965	6968	6971	6974	6977	6980
6984	6990	6997	7003	7009	7023	7026	7029	7032	7035	7047	7050	7053	7056	7059
7062	7065	7069	7075	7082	7088	7094	7105	7138	7141	7145	7148	7151	7154	7157
7169	7172	7175	7178	7181	7184	7187	7191	7197	7204	7210	7216	7230	7233	7236
7239	7242	7254	7257	7260	7263	7266	7269	7272	7276	7282	7289	7295	7301	7312
7343	7346	7349	7352	7355	7358	7361	7364	7367	7373	7376	7379	7382	7385	7388
7391	7394	7397	7400	7403	7406	7412	7424	7427	7430	7433	7436	7440	7449	7461

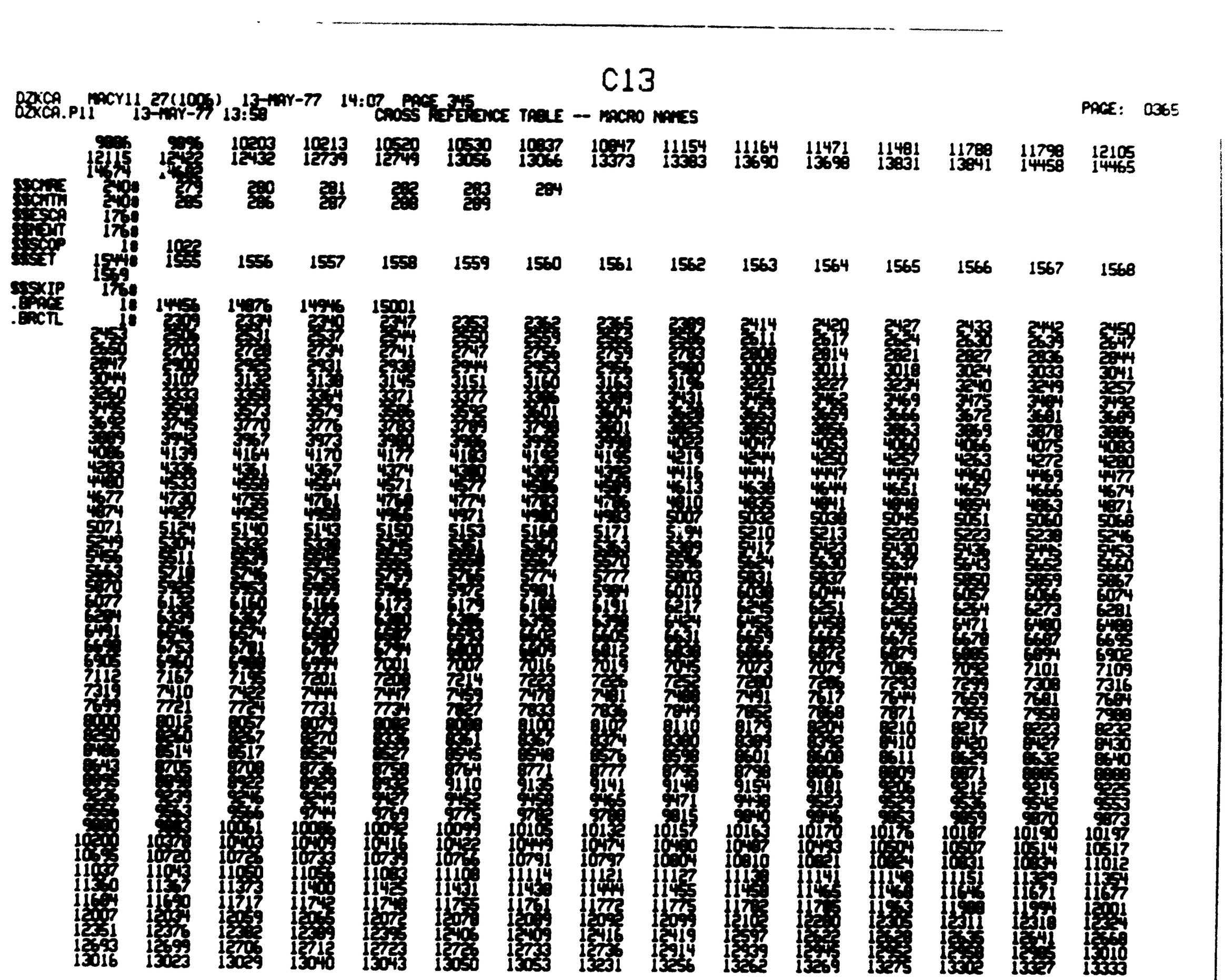


```
        11550  11553  11556  11559  11562  11565  11568  11571  11574  11577  11580  11583  11586  11589  11592
        11595  11598  11601  11604  11607  11610  11613  11616  11619  11623  11627  11633  11640  11651  11654
        11657  11660  11663  11667  11673  11680  11686  11692  11695  11698  11707  11719  11722  11725  11728
        11731  11734  11738  11744  11751  11757  11763  11770  11813  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  11960  11965  11968  11971  11974  11977  11980  11984  11990
        11997  12003  12009  12012  12015  12021  12036  12039  12042  12045  12048  12051  12055  12061  12068
        12074  12080  12095  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  12282  12285  12288  12291  12294  12297  12301  12307  12314  12320  12326  12329  12332
        12341  12363  12366  12369  12372  12375  12378  12381  12384  12387  12391  12397  12412  12447  12450
        12453  12456  12459  12462  12465  12471  12474  12477  12480  12483  12486  12489  12493  12445  12448
        12501  12504  12507  12510  12513  12516  12519  12522  12525  12528  12531  12534  12537  12540  12543
        12546  12549  12552  12555  12558  12561  12564  12567  12570  12574  12578  12581  12598  12602  12606
        12608  12611  12614  12618  12624  12631  12637  12643  12646  12649  12658  12670  12673  12676  12679
        12682  12685  12689  12695  12702  12708  12714  12729  12764  12767  12770  12773  12776  12779  12782
        12788  12791  12794  12797  12800  12803  12806  12809  12812  12815  12818  12821  12824  12827  12830
        12833  12836  12839  12842  12845  12848  12851  12864  12857  12860  12863  12866  12869  12872  12875
        12878  12881  12884  12887  12891  12895  12901  12916  12919  12922  12926  12928  12931  12935  12941
        12948  12954  12960  12963  12966  12975  12987  12990  12993  12996  12999  13002  13006  13012  13019
        13025  13031  13046  13081  13084  13087  13090  13093  13096  13099  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  13208
        13212  13218  13233  13236  13239  13242  13245  13248  13252  13258  13265  13271  13277  13280  13283
        13292  13304  13307  13310  13313  13316  13319  13323  13329  13336  13342  13348  13363  13398  13401
        13404  13407  13410  13413  13416  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  13525  13529  13535  13550  13553  13556
        13559  13562  13565  13571  13575  13582  13585  13588  13591  13594  13597  13604  13621  13624  13627
        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  13900
        13903  13906  13909  13912  13915  13918  13921  13924  13927  13930  13933  13936  13939  13942  13945
        13948  13951  13954  13957  13960  13963  13966  13969  13972  13975  13978  13981  13984  13987  13990
        13993  13996  13999  14002  14005  14008  14011  14014  14017  14020  14023  14026  14030  14033  14045
        14048  14052  14058  14062  14065  14068  14071  14074  14078  14082  14086  14090  14094  14098  14102
        14108  14112  14120  14124  14130  14134  14140  14144  14150  14154  14160  14164  14170  14174  14180
        14184  14190  14194  14203  14206  14210  14216  14219  14222  14225  14229  14232  14236  14242  14246
        14251  14255  14261  14264  14268  14274  14277  14281  14287  14290  14294  14298  14303  14307  14313
        14316  14320  14326  14329  14333  14342  14345  14348  14351  14354  14366  14369  14372  14375  14379
        14407  14416  14419  14431  14434  14437  14440  14443  14447  14460  14463  14466  14469  14492  14495
        14498  14501  14504  14507  14510  14513  14516  14519  14522  14525  14528  14531  14537  14543  14549
        14552  14555  14561  14564  14567  14573  14576  14582  14588  14594  14597  14603  14606  14622  14625
        14637  14640  14643  14646  14649  14653  14663  14688  14701  14704  14707  14710  14713  14716  14722
        14725  14731  14737  14746  14749  14757  14783  14814  14817  14820  14823  14829  14832  14838  14844
        14850  14859  14865  14868  14871  14890  14893  14896  14899  14902  14908  14914  14920  14929  14935
        14938  14941  14954  14957  14960  14963  14966  14972  14978  14984  14987  14993  15009  15012  15015
        15018  15021  15024  15027  15033  15040  15046
MULT    176#
NEWTST  176#
OR        1#  3323  3421  7553  7610  7814  10051  10119  14540  14719  14728  14752  14826  14841  14911
          14975
POP     176#  1220  1221  1378  1694  1695  2249
PUSH    176#  1181  1183  1204  1352  1668  1674  2212
REPORT    1#  176#
```

```
SCOPE      71#  15166
SETDBI    176#
SETDBA    176#
SETTRA   1544#  1555  1556  1557  1558  1559  1560  1561  1562  1563  1564  1565  1566  1567  1568
         1569
SETUP     176#
SHFBRT      1#  2358  2438  2555  2635  2752  2832  2949  3029  3156  3245  3382  3480  3597  3677
         3794  3874  3991  4071  4188  4268  4385  4465  4582  4662  4779  4859  4976  5056  5256
         5441  5563  5648  5770  5855  5977  6062  6184  6269  6391  6476  6598  6683  6805  6890
         7012  7097  7219  7304  7370  7562  7948  8228  8385
SKIP      176#
SLASH     176#
SPACE     176#
STARS     176#   219   242   294   297   483   485   492   961  1028  1097  1176  1235  1241  1270
         1338  1523  1664  1680
SUBMC       1#  11953  12021
SUBI        1#
SUB1C       1#  8541  8625  8701  8791  11319  11387
SUB2C       1#  2305  2385  2502  2582  2699  2779  2896  2976  3103  3192  3329  3427  3544  3624
         3741  3821  3938  4018  4135  4215  4332  4412  4529  4609  4726  4806  4923  5003  5120
         5190  5300  5385  5507  5592  5714  5799  5921  6006  6128  6213  6335  6420  6542  6627
         6749  6834  6956  7041  7163  7248  7418  7455  7655  7695  8053  8175  8246  8332  8406
         8482  8572  8732  9106  9177  9423  9494  9740  9811  10057  10128  10374  10445  10691  10762
        11008  11079  11325  11396  11642  11713  11959  12030  12276  12347  12593  12664  12910  12981  13221
        13227  13289  13298  13544  13615  13979  14360  14425  14612  14631
SWRSU     176#
TRMTRP   1544#
TYPBIN    176#
TYPDEC    176#
TYPNAM    176#
TYPNUM    176#
TYPOCS    176#
TYPOCT    176#
TYPTXT    176#
$ADC        1#  5161  5231  8008  8096  8237  8397  8532  8616  8695  8782  8864  8915  12904  12972
        13791  13967  14769  14772
$ADD        1#  9094  9165  9230  9411  9482  9547  9728  9799  9864  10045  10116  10181  10362  10433
        10498  10679  10685  10750  10753  10815  10996  11067  11132  11313  11384  11449  11630  11701  11766
        11947  12018  12083  12264  12335  12400  12581  12652  12717  12898  12969  13034  13215  13285  13351
        13532  13603  13668  14862  14932
$ALUTO     30#  8811
$ALUT1     30#  8934  9251  9568  9885  10202  10519  10836  11153  11470  11787  12104  12421  12738  13055
        13372
$ASL        1#  12587  12655
$AUTO       1#  817
$BEGIN      1#  2281  2478  2675  2872  3070  3287  3520  3717  3914  4111  4308  4505  4702  4899
         5097  5274  5481  5688  5895  6102  6304  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  14490  14698  14806
$BR         1#  2333  2339  2346  2352  2364  2413  2419  2426  2432  2449  2452  2530  2536  2543
         2549  2561  2610  2616  2623  2629  2646  2649  2727  2733  2740  2746  2758  2807  2813
         2820  2826  2843  2849  2924  2930  2937  3004  3010  3017  3023  3040  3043
         3131  3137  3144  3150  3162  3220  3226  3256  3259  3357  3363  3370  3376
         3388  3455  3461  3468  3474  3491  3494  3585  3591  3603  3652  3658  3665
         3671  3688  3691  3769  3775  3782  3788  3855  3862  3868  3885  3888  3966
         3972  3979  3985  3997  4046  4052  4059  4085  4163  4169  4176  4182  4184
         4243  4249  4256  4262  4279  4282  4360  4379  4391  4440  4446  4453  4459
```

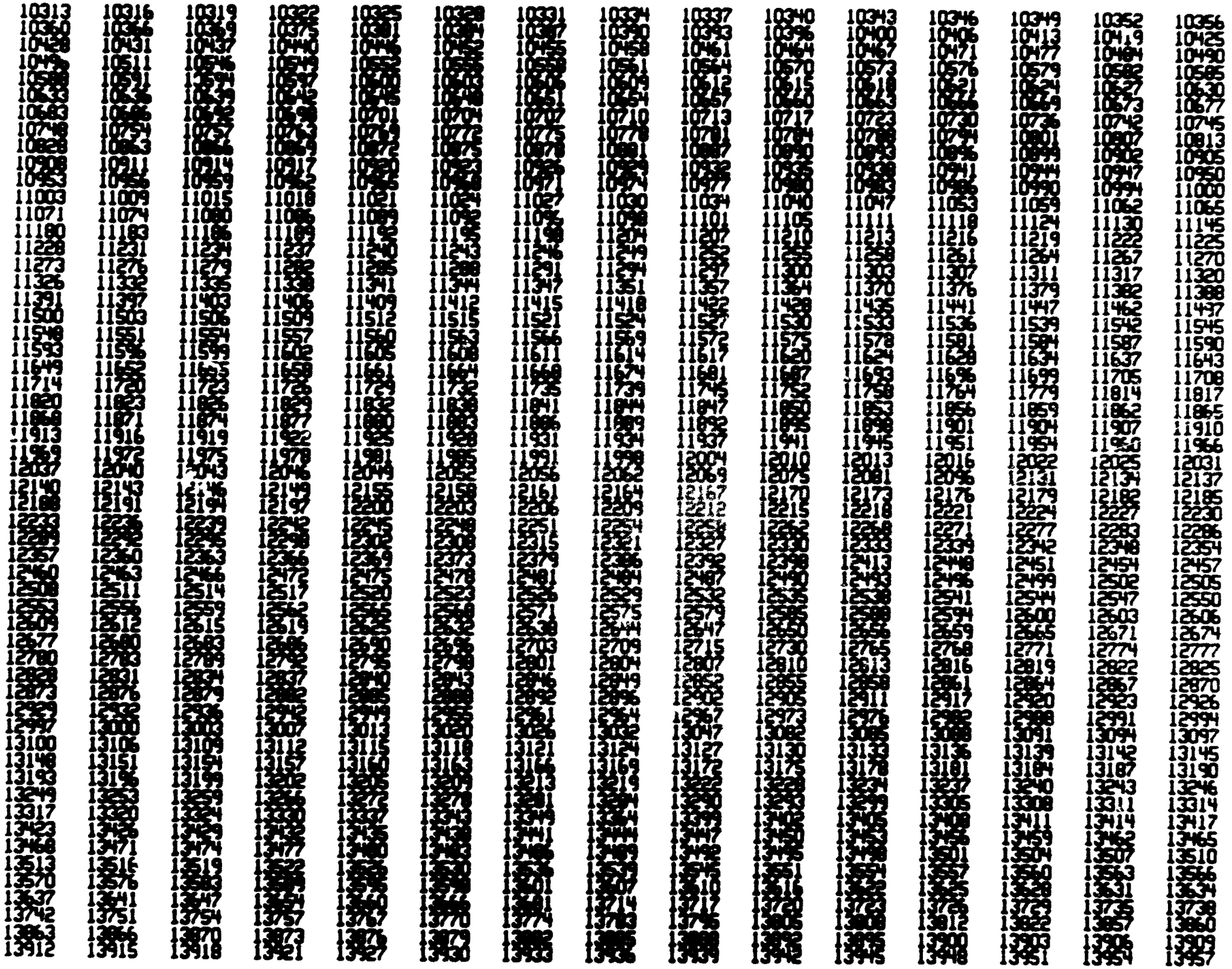
```
                4476  4479  4557  4563  4570  4576  4588  4637  4643  4650  4656  4673  4676  4754  4760
                4767  4773  4785  4834  4840  4847  4853  4870  4873  4951  4957  4964  4970  4982  5031
                5037  5044  5050  5067  5070  5139  5142  5145  5153  5170  5203  5212  5219  5223  5245
                5248  5331  5337  5344  5350  5423  5426  5432  5435  5442  5445  5451  5457  5464  5551
                5557  5563  5623  5630  5636  5643  5650  5657  5663  5730  5737  5744  5750  5757  5836
                5843  5849  5856  5863  5869  5876  5882  5889  5895  5902  5909  5915  5921  5928  6076
                6159  6165  6172  6178  6185  6192  6198  6205  6211  6218  6224  6231  6237  6244  6316
                6397  6451  6457  6464  6470  6477  6483  6490  6496  6503  6509  6516  6522  6529  6671
                6677  6684  6690  6697  6703  6710  6716  6723  6729  6736  6742  6749  6755  6762  6987
                6993  7000  7006  7013  7019  7026  7032  7039  7045  7052  7058  7065  7071  7213  7265
                7272  7278  7285  7291  7298  7304  7311  7317  7324  7330  7337  7343  7350  7356  7720
                7723  7730  7733  7739  7746  7753  7759  7766  7772  7779  7785  7792  7798  7805  8109
                8203  8209  8216  8222  8229  8235  8242  8248  8255  8261  8268  8274  8281  8287  8429
                8513  8516  8523  8530  8536  8543  8549  8556  8562  8569  8575  8582  8588  8595  8762
                8770  8776  8783  8789  8796  8802  8809  8815  8822  8828  8835  8841  8848  9147  9153
                9205  9211  9218  9224  9231  9238  9244  9251  9257  9264  9270  9277  9283  9290  9541
                9555  9562  9568  9575  9581  9588  9594  9601  9607  9614  9620  9627  9633  9640  9646
               10091 10098 10104 10156 10162 10169 10175 10182 10188 10195 10201 10208 10415 10421 10473
               10479 10486 10492 10506 10513 10516 10719 10726 10732 10739 10745 10752 10758 10765 10823
               10830 10833 11036 11042 11049 11055 11107 11113 11120 11126 11133 11140 11147 11353 11359
               11366 11372 11424 11430 11437 11443 11457 11464 11470 11476 11483 11489 11496 11741 11747
               11754 11760 11774 11781 11787 11987 11993 12000 12006 12013 12019 12026 12032 12077 12091
               12101 12304 12310 12317 12323 12375 12381 12388 12394 12401 12407 12414 12420 12427 12434
               12640 12692 12698 12705 12711 12725 12732 12735 12745 12752 12758 12765 12771 12778 13015
               13028 13042 13049 13052 13255 13261 13268 13274 13326 13332 13339 13345 13352 13359 13362
               13572 13578 13585 13591 13643 13649 13656 13662 13676 13683 13689 13696 13702 13709 13716
               13779 13800 13814 13817 13824 13827 13985 13992 13998 14017 14023 14030 14036 14043 14075
               14086 14095 14105 14115 14127 14137 14147 14157 14167 14177 14187 14197 14213 14232 14245
               14258 14271 14284 14297 14310 14323 14336 14382 14385 14394 14403 14410 14413 14450 14454
               14585 14600 14618 14656 14659 14666 14669 14763 14778 14786 14874 14944 14999 15043 15049
SBRFLT      9#  5104  5174
SBUFFE      1#  1781
SCOMP       1#
SCYCLE      1#  1852
SDEC        1#  7951  7996  8084  8900  8912  8981  9232  9298  9549  9615  9866  9932 10183 10249
           10500 10566 10817 10883 11134 11200 11451 11517 11768 11834 12085 12151 12402 12468 12719
           12785 13036 13102 13353 13419 13538 13606 13670 14002 14036 14397
SEOP        1#   953
SERTBL      1#   413
SFINI       1# 15311
SFLOT       3#  2280  2368  2485  2565  2682  2762  2879  2959  3077  3166  3294  3392  3527  3607
           3724  3804  3921  4001  4118  4198  4315  4395  4512  4592  4709  4789  4906  4986
SFLTI       4#  2258  2455  2652  2849  3046  3262  3497  3694  3891  4088  4285  4482  4679  4876
SGETPA      1#
SHEADE      1#    42
SHLRSR     30# 14457
SIFEQ       1#  2302  2382  2499  2579  2696  2776  2893  2973  3100  3180  3326  3424  3541  3621
           3738  3818  3935  4015  4132  4212  4329  4409  4526  4606  4723  4803  4920  5000  5117
           5187  5297  5382  5504  5589  5711  5796  5918  6003  6125  6210  6332  6417  6539  6624
           6746  6831  6953  7038  7160  7245  7415  7452  7652  7692  8050  8172  8329  8479  8569
           8729  9103  9174  9420  9491  9737  9808 10054 10125 10371 10442 10688 10759 11005 11076
          11322 11393 11639 11710 11956 12027 12273 12344 12590 12661 12907 12978 13224 13295 13541
          13612 14357 14422 14628
SIFHI       1#  8538  8622  8698  8788
SIFKIS      1# 13976
SIFLO       1#  8243  8403
```

```
$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
$LOCLK     29#
$MACRO      1#
$MEM10     21#  8115
$MEM11     19#  8272
$MEM12     22#  8432
$MEM31     23#  8645
$MFLT      18#  8137  8294
$MOCK       1#
$MSG        1#  1769
$NOISE     30# 13830
$NPRG      29#
$OPD1      25#  7321
$OPD2      26#  7494
$OPD4      27#  7873
$OPS        8#  7737
$OUT        1#   967
$PC         1#  1659
$PSECT     29# 13689
$QUEUE      6#
$RAD50      9#  5073
$REL       30# 14673
$QUEST      1#  1967  1979  1987  2044  2052
$RADMCL     1#  1713
$RCLK       1#  1716  1719  1756  1761
$ROL        1#  5158  5228 11002 11070
$ROVAR      1#   363
$SCAD0      1#  1035
$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#
$TSTN       1#  2269  2466  2663  2860  3058  3275  3508  3705  3902  4099  4296  4493  4690  4887
            5085  5262  5469  5676  5883  6090  6297  6504  6711  6918  7125  7331  7505  7748  7885
            8125  8282  8443  8656  8822  8948  9265  9582  9899 10216 10533 10850 11167 11484 11801
           12118 12435 12752 13069 13386 13701 13844 14468 14686
$UPADD      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
```



13340	13346	13357	13360	13367	13370	13548	13573	13579	13586	13592	13619	13644	13650	13657	
13663	13674	13677	13681	13687	13776	13735	13738	13760	13763	13777	13780	13786	13798	13801	
13815	13818	13825	13838	13845	13916	13935	13938	13960	13965	14016	14024	14040	14056	14066	
14076	14086	14096	14116	14116	14128	14136	14156	14157	14163	14176	14188	14198	14214	14233	
14246	14259	14272	14286	14296	14311	14326	14327	14341	14343	14356	14362	14376	14401	14404	
14411	14414	14427	14431	14447	14447	14461	14471	14474	14477	14496	14601	14616	14619	14635	
14657	14660	14667	14673	14706	14741	14761	14771	14776	14777	14796	14848	14854	14857	14875	
[illegible]	14924	14927	14945	14982	14997	15000	15021	15027	15044	15050					
18	2283	2286	2290	2294	2297	2300	2312	2315	2318	2321	2324	2327	2331		
2337	2344	2350	2354	2357	2374	2377	2380	2392	2395	2398	2401	2404	2407		
		2424	2430	2434	2447	2460	2463	2471	2474	2477	2497	2503	2509	2512	
		2521	2527	2531	2534	2541	2547	2557	2560	2563	2574	2577	2583	2589	
		2538	2601	2604	2608	2614	2621	2624	2643	2644	2677	2680	2684	2688	
	2654	2700	2706	2709	2712	2715	2718	2721	2725	2731	2738	2744	2750	2764	
2768	2771	2774	2780	2786	2790	2793	2796	2799	2802	2805	2811	2818	2824	2830	
2841	2874	2877	2881	2885	2888	2891	2894	2897	2900	2905	2912	2915	2918	2922	
2928	2935	2941	2947	2951	2955	2958	2961	2964	2967	2980	2983	2986	2990	2998	
3002	3008	3015	3021	3027	3030	3034	3037	3040	3043	3046	3049	3052	3056	3098	
3104	3110	3113	3116	3119	3122	3125	3128	3131	3134	3137	3142	3148	3154	3168	3172
3178	3181	3184	3187	3193	3196	3199	3202	3205	3208	3211	3214	3218	3224	3231	3237
3243	3254	3289	3292	3296	3300	3303	3306	3309	3312	3315	3318	3321	3324	3330	
3336	3339	3342	3345	3348	3351	3354	3357	3360	3374	3380	3394	3398	3401	3404	
3407	3410	3413	3416	3419	3422	3425	3428	3431	3440	3443	3446	3449	3453	3459	
3466	3472	3478	3482	3486	3490	3493	3496	3499	3502	3505	3539	3545	3551	3554	3557
3563	3566	3570	3576	3580	3583	3586	3589	3592	3595	3598	3616	3619	3623	3631	3634
3640	3643	3646	3650	3656	3660	3663	3666	3669	3672	3675	3719	3722	3726	3730	3733
3742	3748	3751	3754	3757	3760	3763	3767	3773	3780	3786	3792	3806	3810	3813	
3816	3822	3828	3831	3834	3837	3840	3843	3847	3850	3856	3860	3872	3883	3916	
3919	3923	3927	3930	3933	3939	3945	3948	3951	3954	3957	3960	3964	3970	3977	
3983	3989	4003	4007	4010	4013	4019	4025	4028	4031	4034	4037	4040	4044	4050	
4057	4063	4069	4080	4113	4116	4120	4124	4127	4130	4136	4142	4145	4148	4151	
4154	4157	4161	4167	4174	4180	4186	4200	4204	4207	4210	4216	4222	4225	4238	
4231	4234	4237	4241	4247	4254	4260	4266	4277	4310	4313	4317	4321	4324	4327	
4333	4339	4342	4345	4348	4351	4354	4358	4364	4371	4377	4383	4397	4401	4454	
4407	4413	4419	4422	4425	4428	4431	4434	4438	4444	4451	4457	4463	4474	4507	
4510	4514	4518	4521	4524	4530	4536	4539	4542	4545	4548	4551	4555	4561	4568	
4574	4580	4594	4598	4601	4604	4610	4616	4619	4622	4625	4628	4631	4635	4641	
4648	4654	4660	4671	4704	4707	4711	4715	4718	4721	4727	4733	4736	4739	4742	
4745	4748	4752	4758	4765	4771	4777	4791	4795	4798	4801	4807	4813	4816	4819	
4822	4825	4828	4832	4838	4845	4851	4857	4868	4901	4904	4908	4912	4915	4918	
4921	4930	4933	4936	4939	4942	4945	4949	4965	4968	4969	4972	4988	4992	4995	
4998	5004	5010	5013	5016	5019	5022	5025	5029	5035	5042	5046	5054	5065	5068	
5102	5106	5109	5112	5115	5121	5127	5130	5133	5137	5147	5156	5165	5176	5183	
5186	5189	5191	5197	5200	5203	5207	5217	5226	5236	5243	5276	5278	5283	5342	
5296	5299	5305	5308	5307	5310	5313	5316	5319	5322	5325	5328	5334	5337	5340	
5343	5347	5371	5374	5377	5380	5386	5392	5395	5398	5401	5407	5407	5410	5414	
5427	5433	5439	5450	5483	5486	5490	5493	5496	5498	5502	5508	5514	5517		
5529	5532	5536	5542	5549	5555	5561	5575	5578	5581	5587					
5597	5602	5605	5608	5611	5614	5617	5621	5624	5627	5634	5640	5646	5657	5660	
5697	5700	5703	5706	5709	5715	5721	5724	5727	5730	5733	5739	5743			
5762	5768	5782	5785	5788	5791	5794	5800	5803	5806	5809	5812	5815	5818		
5828	5834	5841	5847	5853	5864	5897	5900	5907	5910	5913	5916				
5931	5934	5937	5940	5943	5946	5950	5956	5963	5969	5975	5982				
6001	6007	6013	6016	6019	6022	6025	6028	6031	6035	6041	6048	6054			
6071	6104	6107	6111	6114	6117	6120	6123	6129	6135	6138	6141	6144	6147		
6150	6153	6157	6163	6170	6176	6182	6196	6199	6202	6205	6208	6214	6220	6223	

6226	6229	6232	6235	6238	6242	6248	6255	6261	6267	6278	6311	6314	6318	6321
6327	6330	6333	6336	6342	6345	6348	6351	6354	6357	6360	6364	6370	6377	6383
6403	6406	6409	6412	6415	6421	6427	6430	6433	6436	6439	6442	6445	6448
6474	6477	6480	6483	6487	6490	6493	6496	6499	6502	6505	6508	6511	6514	6517	6520
6524	6527	6530	6533	6537	6540	6543	6546	6549	6552	6555	6558	6561	6564	6567	6570
6574	6577	6580	6583	6587	6590	6593	6596	6599	6602	6605	6608	6611	6614	6617	6620
6624	6627	6630	6633	6637	6640	6643	6646	6649	6652	6655	6658	6661	6664	6667	6670
6674	6677	6680	6683	6687	6690	6693	6696	6699	6702	6705	6708	6711	6714	6717	6720
6724	6727	6730	6733	6737	6740	6743	6746	6749	6752	6755	6758	6761	6764	6767	6770
6774	6777	6780	6783	6787	6790	6793	6796	6799	6802	6805	6808	6811	6814	6817	6820
6824	6827	6830	6833	6837	6840	6843	6846	6849	6852	6855	6858	6861	6864	6867	6870
6874	6877	6880	6883	6887	6890	6893	6896	6899	6902	6905	6908	6911	6914	6917	6920
6924	6927	6930	6933	6937	6940	6943	6946	6949	6952	6955	6958	6961	6964	6967	6970
6974	6977	6980	6983	6987	6990	6993	6996	6999	7002	7005	7008	7011	7014	7017	7020
7024	7027	7030	7033	7037	7040	7043	7046	7049	7052	7055	7058	7061	7064	7067	7070
7074	7077	7080	7083	7087	7090	7093	7096	7099	7102	7105	7108	7111	7114	7117	7120
7124	7127	7130	7133	7137	7140	7143	7146	7149	7152	7155	7158	7161	7164	7167	7170
7174	7177	7180	7183	7187	7190	7193	7196	7199	7202	7205	7208	7211	7214	7217	7220
7224	7227	7230	7233	7237	7240	7243	7246	7249	7252	7255	7258	7261	7264	7267	7270
7274	7277	7280	7283	7287	7290	7293	7296	7299	7302	7305	7308	7311	7314	7317	7320
7324	7327	7330	7333	7337	7340	7343	7346	7349	7352	7355	7358	7361	7364	7367	7370
7374	7377	7380	7383	7387	7390	7393	7396	7399	7402	7405	7408	7411	7414	7417	7420
7424	7427	7430	7433	7437	7440	7443	7446	7449	7452	7455	7458	7462	7465	7468
7474	7477	7480	7483	7487	7490	7493	7496	7499	7502	7505	7508	7511	7514	7517	7520
7524	7527	7530	7533	7537	7540	7543	7546	7549	7552	7555	7558	7561	7564	7567	7570
7574	7577	7580	7583	7587	7590	7593	7596	7599	7602	7605	7608	7611	7614	7617	7620
7624	7627	7630	7633	7637	7640	7643	7646	7649	7652	7655	7658	7661	7664	7667	7670
7674	7677	7680	7683	7687	7690	7693	7696	7699	7702	7705	7708	7711	7714	7717	7720
7724	7727	7730	7733	7737	7740	7743	7746	7749	7752	7755	7758	7761	7764	7767	7770
7774	7777	7780	7783	7787	7790	7793	7796	7799	7802	7805	7808	7811	7814	7817	7820
7824	7827	7830	7833	7837	7840	7843	7846	7849	7852	7855	7858	7861	7865
7874	7877	7880	7883	7887	7890	7893	7896	7899	7902	7905	7908	7911	7914	7917	7920
7924	7927	7930	7933	7937	7940	7943	7946	7949	7952	7955	7958	7961	7964	7967	7970
7974	7977	8000	8003	8007	8010	8013	8016	8019	8022	8025	8028	8031	8034	8037	8040
8044	8047	8050	8053	8057	8060	8063	8066	8069
8074	8077	8100	8103	8107	8110	8113	8116	8119	8122	8125	8128	8131	8134	8137	8140
8144	8147	8150	8153	8157	8160	8163	8166	8169	8172	8175	8178	8181	8184	8187	8190
8194	8197	8200	8203	8207	8210	8213	8216	8219	8222	8225	8228	8231	8234	8237	8240
8244	8247	8250	8253	8257	8260	8263	8266	8269	8272	8275	8278	8281	8284	8287	8290
8294	8297	8300	8303	8307	8310	8313	8316	8319	8322	8325	8328	8331	8334	8337	8340
8344	8347	8350	8353	8357	8360	8363	8366	8369	8372	8375	8378	8381	8384	8387	8390
8394	8397	8400	8403	8407	8410	8413	8416	8419	8422	8425	8428	8431	8434	8437	8440
8444	8447	8450	8453	8457	8460	8463	8466	8469	8472	8475	8478	8481	8484	8487	8490
8494	8497	8500	8503	8507	8510	8513	8516	8519	8522	8525	8528	8531	8534	8537	8540
8544	8547	8550	8553	8557	8560	8563	8566	8569	8572	8575	8578	8581	8584	8587	8590
8594	8597	8600	8603	8607	8610	8613	8616	8619	8622	8625	8628	8631	8634	8637	8640
8644	8647	8650	8653	8657	8660	8663	8666	8669	8672	8675	8678	8681	8684	8687	8690
8694	8697	8700	8703	8707	8710	8713	8716	8719	8722	8725	8728	8731	8734	8737	8740
8744	8747	8750	8753	8757	8760	8763	8766	8769	8772	8775	8778	8781	8784	8787	8790
8794	8797	8800	8803	8807	8810	8813	8816	8819	8822	8825	8828	8831	8834	8837	8840
8844	8847	8850	8853	8857	8860	8863	8866	8869	8872	8875	8878	8881	8884	8887	8890
8894	8897	8900	8903	8907	8910	8913	8916	8919	8922	8925	8928	8931	8934	8937	8940
8944	8947	8950	8953	8957	8960	8963	8966	8969	8972	8975	8978	8981	8984	8987	8990
8994	8997	9000	9003	9007	9010	9013	9016	9019	9022	9025	9028	9031	9034	9037	9040
9044	9047	9050	9053	9057	9060	9063	9066	9069	9072	9075	9078	9081	9084	9087	9090
9094	9097	9100	9103	9107	9110	9113	9116	9119	9122	9125	9128	9131	9134	9137	9140
9144	9147	9150	9153	9157	9160	9163	9166	9169	9172	9175	9178	9181	9184	9187	9190
9194	9197	9200	9203	9207	9210	9213	9216	9219	9222	9225	9228	9231	9234	9237	9240
9244	9247	9250	9253	9257	9260	9263	9266	9269	9272	9275	9278	9281	9284	9287	9290
9294	9297	9300	9303	9307	9310	9313	9316	9319	9322	9325	9328	9331	9334	9337	9340
9344	9347	9350	9353	9357	9360	9363	9366	9369	9372	9375	9378	9381	9384	9387	9390
9394	9397	9400	9403	9407	9410	9413	9416	9419	9422	9425	9428	9431	9434	9437	9440
9444	9447	9450	9453	9457	9460	9463	9466	9469	9472	9475	9478	9481	9484	9487	9490
9494	9497	9500	9503	9507	9510	9513	9516	9519	9522	9525	9528	9531	9534	9537	9540
9544	9547	9550	9553	9557	9560	9563	9566	9569	9572	9575	9578	9581	9584	9587	9590
9594	9597	9600	9603	9607	9610	9613	9616	9619	9622	9625	9628	9631	9634	9637	9640
9644	9647	9650	9653	9657	9660	9663	9666	9669	9672	9675	9678	9681	9684	9687	9690
9694	9697	9700	9703	9707	9710	9713	9716	9719	9722	9725	9728	9731	9734	9737	9740
9744	9747	9750	9753	9757	9760	9763	9766	9769	9772	9775	9778	9781	9784	9787	9790
9794	9797	9800	9803	9807	9810	9813	9816	9819	9822	9825	9828	9831	9834	9837	9840
9844	9847	9850	9853	9857	9860	9863	9866	9869	9872	9875	9878	9881	9884	9887	9890
9894	9897	9900	9903	9907	9910	9913	9916	9919	9922	9925	9928	9931	9934	9937	9940
9944	9947	9950	9953	9957	9960	9963	9966	9969	9972	9975	9978	9981	9984	9987	9990
9994	9997	10002	10005	10008	10011	10014	10017	10020	10023	10026	10029	10032	10035
10039	10043	10049	10052	10058	10064	10067	10070	10073	10076	10079	10083	10096	10102
10108	10111	10114	10120	10123	10129	10135	10138	10141	10144	10147	10150	10154	10160	10167
10173	10179	10194	10229	10232	10235	10238	10241	10244	10247	10253	10256	10259	10262	10265
10268	10271	10274	10277	10280	10283	10286	10289	10292	10295	10298	10301	10304	10307	10310



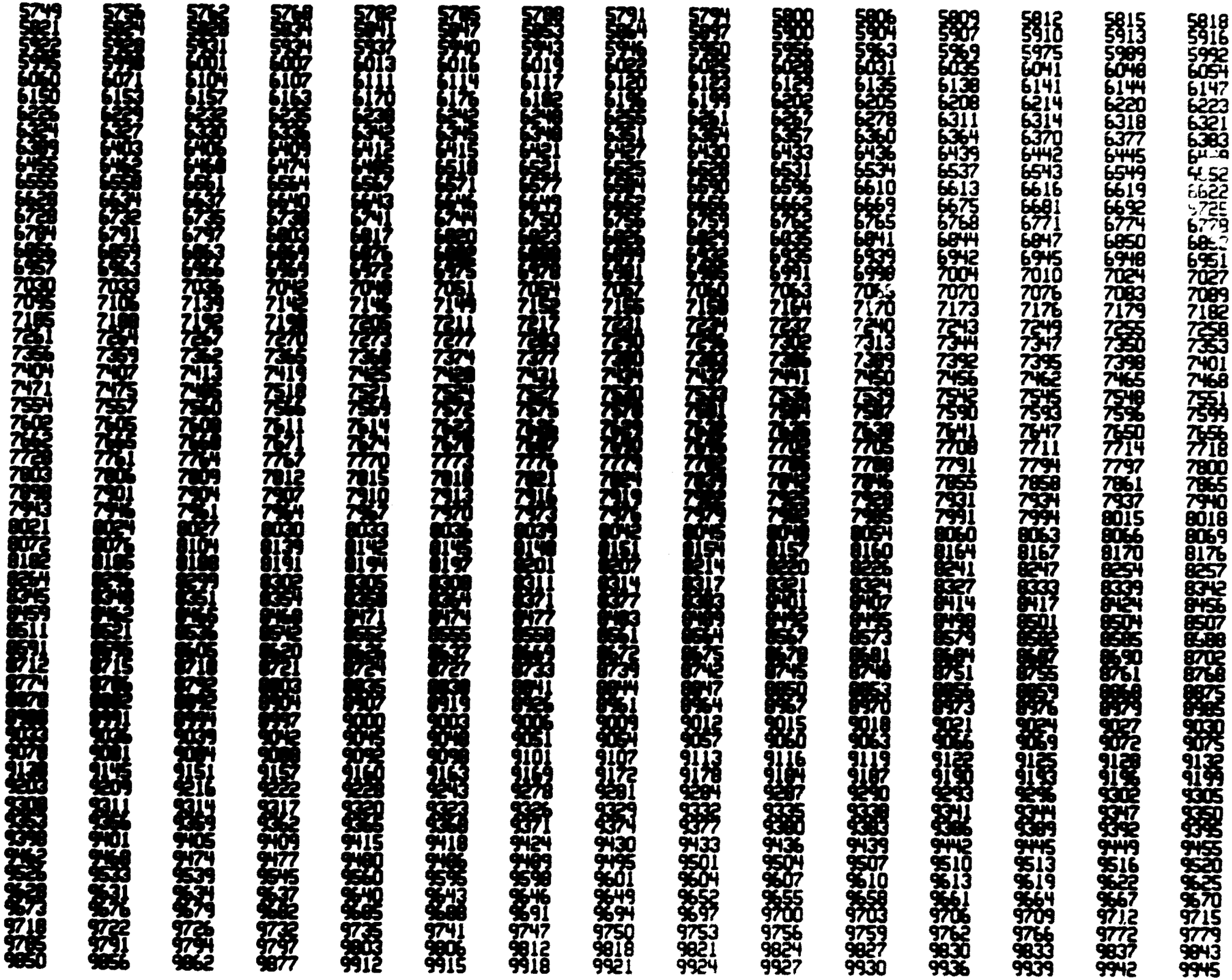
```
                13962  13971  13974  13980  13990  13996  14009  14021  14027  14031  14034  14046  14049  14053  14059
                14063  14069  14073  14079  14083  14089  14093  14099  14103  14109  14113  14121  14125  14131  14135
                14141  14145  14151  14155  14161  14165  14171  14175  14181  14185  14191  14195  14204  14207  14211
                14217  14220  14223  14226  14230  14236  14239  14243  14249  14252  14256  14262  14265  14269  14275
                14278  14282  14288  14291  14295  14301  14304  14308  14314  14317  14321  14327  14330  14334  14343
                14346  14349  14352  14355  14361  14367  14370  14373  14376  14380  14408  14417  14420  14426  14432
                14435  14438  14441  14444  14448  14481  14484  14497  14490  14493  14496  14499  14502  14505  14508
                14511  14514  14517  14520  14523  14526  14529  14532  14535  14538  14541  14544  14550  14553  14556
                14562  14565  14568  14574  14580  14583  14589  14595  14598  14604  14607  14613  14623  14626  14632
                14638  14641  14644  14647  14650  14654  14664  14699  14702  14705  14708  14711  14714  14717  14720
                14723  14726  14729  14732  14738  14747  14750  14753  14758  14784  14815  14818  14821  14824  14827
                14830  14833  14836  14839  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
.DESTM      18  2283   2286   2290   2294   2297   2300   2312   2315   2318   2321   2324   2327   2331   2337
            2344   2350   2356   2370   2374   2377   2380   2342   2345   2398   2401   2404   2407   2411   2417
            2424   2430   2436   2447   2480   2483   2487   2491   2494   2497   2503   2512   2515   2518   2521
            2524   2528   2534   2541   2547   2553   2567   2571   2574   2577   2583   2592   2595   2598   2601
            2604   2608   2614   2621   2627   2633   2644   2677   2680   2684   2688   2691   2694   2706   2709
            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   3191   3202   3205   3208   3211   3214   3218
            3224   3231   3237   3243   3254   3289   3292   3295   3298   3303   3306   3312   3315   3318   3321
            3336   3339   3342   3345   3348   3351   3354   3357   3360   3374   3380   3394   3398   3401   3404
            3410   3413   3416   3419   3434   3437   3440   3443   3446   3449   3453   3459   3466   3472   3478
            3489   3522   3525   3529   3533   3537   3540   3543   3546   3557   3560   3563   3566   3570   3576
            3583   3589   3595   3609   3613   3617   3620   3623   3626   3637   3640   3643   3646   3650   3656
            3663   3669   3675   3686   3719   3723   3726   3729   3732   3736   3748   3751   3754   3757   3760
            3767   3773   3780   3796   3799   3802   3805   3808   3811   3815   3828   3831   3834   3837   3840
            3847   3853   3860   3866   3869   3872   3875   3878   3881   3923   3927   3930   3933   3945   3948
            3954   3957   3960   3964   3967   3970   3973   3976   3979   4003   4007   4010   4013   4025   4028
            4034   4037   4040   4044   4047   4050   4053   4056   4059   4080   4113   4116   4120   4124   4127
            4141   4145   4148   4151   4154   4157   4160   4163   4166   4174   4180   4186   4200   4204   4207
            4221   4225   4228   4231   4234   4237   4240   4243   4246   4260   4266   4277   4310   4313
            4318   4321   4324   4327   4330   4333   4336   4339   4342   4358   4364   4371   4377   4383
            4401   4404   4407   4411   4414   4417   4420   4423   4426   4438   4444   4451   4457   4463
            4510   4514   4517   4520   4523   4526   4529   4532   4535   4545   4548   4551   4555   4561
            4577   4580   4584   4587   4590   4594   4597   4600   4603   4625   4628   4631   4635   4641
            4665   4668   4671   4674   4677   4680   4683   4686   4689   4733   4736   4739   4742   4745
            4758   4761   4765   4768   4771   4774   4777   4780   4783   4813   4816   4819   4822   4825
            4838   4841   4845   4848   4851   4854   4857   4860   4863   4912   4915   4916   4930   4933
            4942   4945   4949   4952   4955   4958   4961   4964   4967   4988   4992   4998   5010   5013
            5022   5025   5029   5032   5035   5038   5041   5044   5047   5065   5099   5106   5109   5112
            5130   5133   5137   5140   5143   5147   5150   5153   5156   5179   5182   5197   5200   5203
            5226   5235   5243   5246   5249   5276   5279   5282   5286   5289   5292   5307   5310   5313
            5322   5325   5329   5332   5335   5342   5345   5348   5354   5368   5371   5377   5380   5382
            5401   5404   5407   5410   5414   5417   5420   5423   5427   5433   5439   5483   5486   5490
            5498   5502   5514   5517   5520   5523   5526   5529   5532   5536   5542   5549   5555
            5539   5578   5581   5584   5587   5599   5602   5605   5608   5611   5617   5621   5627
            5575   5640   5646   5657   5690   5693   5697   5700   5703   5706   5709   5714   5721   5724   5727   5730
            5733   5736   5739   5743   5749   5756   5762   5768   5782   5785   5788   5791   5794   5806   5809
            5812   5815   5818   5821   5824   5828   5834   5841   5847   5853   5864   5897   5900   5904   5907
```

5910	5913	5916	5920	5931	5934	5937	5940	5943	5946	5950	5956	5963	5969	5975
5985	5992	5995	5998	6001	6013	6016	6019	6022	6025	6028	6031	6035	6041	6046
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	6205	6208	6220	6223	6226
6253	6280	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	6482	6518	6521	6525	6528	6531	6534	6537	6549	6552	6555	6558	6561	6564
6567	6571	6577	6584	6590	6596	6610	6613	6616	6619	6622	6634	6637	6640	6643
6646	6649	6652	6656	6662	6669	6675	6681	6692	6725	6728	6732	6735	6738	6741
6744	6756	6759	6762	6765	6768	6771	6774	6778	6784	6791	6797	6803	6817	6820
6823	6826	6829	6841	6844	6847	6850	6853	6856	6859	6863	6869	6876	6882	6888
6898	6932	6935	6939	6942	6945	6948	6951	6963	6966	6969	6972	6975	6978	6981
6985	6991	6998	7004	7010	7024	7027	7030	7033	7036	7048	7051	7054	7057	7060
7063	7066	7070	7076	7063	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	7623	7626	7629	7632	7635	7638	7641	7647	7650	7662
7665	7668	7671	7674	7678	7687	7690	7702	7705	7708	7711	7714	7718	7728	7761
7764	7767	7770	7773	7776	7779	7782	7785	7788	7791	7794	7797	7800	7803	7806
7809	7812	7818	7821	7824	7839	7842	7846	7855	7858	7865	7898	7901	7904	7907
7910	7913	7916	7919	7922	7925	7928	7931	7934	7937	7940	7943	7946	7961	7964
7967	7970	7973	7976	7979	7982	7985	7991	7994	8015	8018	8021	8024	8027	8030
8033	8036	8039	8042	8045	8048	8060	8063	8066	8069	8072	8076	8104	8139	8142
8145	8148	8151	8154	8157	8160	8164	8167	8170	8182	8185	8188	8191	8194	8197
8201	8207	8214	8220	8226	8241	8254	8257	8264	8296	8299	8302	8305	8308	8311
8314	8317	8321	8324	8327	8339	8342	8345	8348	8351	8354	8358	8364	8371	8377
8383	8401	8414	8417	8424	8456	8459	8462	8465	8468	8471	8474	8477	8489	8492
8495	8498	8501	8504	8507	8511	8521	8536	8552	8555	8558	8561	8564	8567	8579
8582	8585	8588	8591	8595	8605	8620	8637	8669	8672	8675	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	8979	8985	8988
8991	8994	8997	9000	9003	9006	9009	9012	9015	9018	9021	9024	9027	9030	9033
9036	9039	9042	9045	9048	9051	9054	9057	9060	9063	9066	9069	9072	9075	9078
9081	9084	9088	9092	9098	9101	9113	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	9276	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

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	10366	10381	10384	10387	10390	10393	10396	10400	10406	10413	10419	10425	10428	10431
10440	10453	10455	10458	10461	10464	10467	10471	10477	10481	10440	10442	10511	10516	10549
10552	10555	10558	10561	10564	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	10673	10677	10683	10698	10701	10704
10707	10710	10713	10717	10723	10730	10736	10742	10745	10748	10757	10769	10772	10775	10778
10791	10784	10788	10794	10801	10807	10813	10820	10863	10866	10869	10872	10875	10878	10881
10867	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
7	10980	10983	10986	10990	10994	11000	11015	11018	11021	11024	11027	11030	11034	11040
7	11053	11059	11062	11065	11074	11086	11089	11092	11095	11098	11101	11105	11111	11118
4	11130	11145	11180	11183	11186	11189	11192	11195	11198	11204	11207	11210	11213	11216
11219	11222	11225	11228	11231	11234	11237	11240	11243	11246	11249	11252	11255	11258	11261
11264	11267	11270	11273	11276	11279	11282	11285	11288	11291	11294	11297	11300	11303	11307
11311	11317	11332	11336	11338	11341	11344	11347	11351	11357	11364	11370	11376	11379	11382
11391	11403	11406	11409	11412	11415	11418	11422	11428	11435	11441	11447	11462	11497	11500
11503	11506	11509	11512	11515	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	11624	11628	11634	11643	11652	11655
11658	11661	11664	11668	11674	11681	11687	11693	11696	11699	11708	11720	11723	11726	11729
11732	11735	11739	11745	11752	11758	11774	11777	11814	11817	11820	11823	11826	11829	11832
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	11941	11955	11961	11966	11969	11972	11975	11978	11981	11985	11991
11998	12004	12010	12013	12016	12025	12037	12040	12043	12046	12049	12052	12056	12062	12069
12075	12081	12096	12131	12134	12137	12140	12143	12146	12149	12155	12158	12161	12164	12167
12170	12173	12176	12179	12182	12185	12188	12191	12194	12197	12200	12203	12206	12209	12212
12215	12218	12221	12224	12227	12230	12233	12236	12239	12242	12245	12248	12251	12254	12258
12262	12268	12283	12286	12289	12292	12295	12298	12302	12308	12315	12321	12327	12330	12333
12342	12354	12357	12360	12363	12366	12369	12373	12379	12386	12392	12398	12413	12448	12451
12454	12457	12460	12463	12466	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	12575	12579	12585	12600	12603	12606
12609	12612	12615	12619	12625	12632	12638	12644	12647	12650	12659	12671	12674	12677	12680
12683	12686	12690	12696	12703	12709	12715	12730	12765	12768	12771	12774	12777	12780	12783
12786	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	12892	12896	12902	12917	12920	12923	12926	12929	12932	12936	12942
12949	12955	12961	12964	12967	12976	12988	12991	12994	12997	13000	13003	13007	13013	13020
13026	13032	13047	13082	13085	13088	13091	13094	13097	13100	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	13209
13213	13219	13234	13237	13240	13243	13246	13249	13253	13259	13266	13272	13278	13281	13284
13293	13305	13308	13311	13314	13317	13320	13324	13330	13337	13343	13349	13364	13399	13402
13405	13408	13411	13414	13417	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	13526	13530	13536	13551	13554	13557
13560	13563	13566	13570	13576	13583	13589	13595	13598	13601	13610	13622	13625	13628	13631
13634	13637	13641	13647	13654	13660	13666	13681	13714	13717	13720	13723	13726	13729	13735
13738	13742	13751	13754	13757	13767	13770	13774	13783	13795	13805	13808	13812	13822	13857
13860	13863	13866	13870	13873	13876	13879	13882	13885	13888	13892	13895	13900	13903	13906
13909	13912	13915	13918	13921	13927	13930	13933	13936	13939	13942	13945	13948	13951	13954
13957	13962	13971	13974	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 14252 14256 14262 14265 14269 14275
14278 14282 14288 14291 14295 14301 14304 14308 14314 14317 14321 14327 14330 14334 14343
14346 14349 14352 14355 14367 14370 14373 14376 14380 14408 14417 14420 14432 14435 14438
14441 14444 14448 14481 14484 14487 14490 14493 14496 14499 14502 14505 14508 14511 14514
14517 14520 14523 14526 14529 14532 14538 14544 14550 14553 14556 14562 14565 14568 14574
14580 14583 14589 14595 14598 14604 14607 14623 14626 14638 14641 14644 14647 14650 14654
14664 14699 14702 14705 14708 14711 14714 14717 14723 14726 14732 14738 14747 14750 14758
14784 14815 14818 14821 14824 14830 14833 14839 14845 14851 14860 14866 14869 14872 14891
14894 14897 14900 14903 14909 14915 14921 14930 14936 14939 14942 14955 14958 14961 14964
14967 14973 14979 14985 14988 14994 15010 15013 15016 15019 15022 15025 15028 15034 15041
15047

.EQUAT 1# 66
.HEADE 1# 31
.MCTL 1# 2283 2286 2290 2294 2297 2300 2306 2312 2315 2318 2321 2324 2327 2331
2337 2344 2350 2356 2370 2374 2377 2380 2386 2392 2395 2398 2401 2404 2407
2411 2417 2424 2430 2436 2447 2480 2483 2487 2491 2494 2497 2503 2509 2512
2515 2518 2521 2524 2528 2534 2541 2547 2553 2567 2571 2574 2577 2583 2588
2592 2595 2598 2601 2604 2608 2614 2621 2627 2633 2644 2677 2680 2684 2688
2691 2694 2700 2706 2709 2712 2715 2718 2721 2725 2731 2738 2744 2750 2764
2768 2771 2774 2780 2786 2789 2792 2795 2798 2801 2805 2811 2818 2824 2830
2841 2874 2877 2881 2885 2888 2891 2897 2903 2906 2909 2912 2915 2918 2922
2928 2936 2941 2947 2961 2965 2968 2971 2977 2983 2986 2989 2992 2995 2998
3002 3008 3015 3021 3027 3038 3072 3075 3079 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 3196 3202 3205 3208 3211 3214 3218 3224 3231 3237
3243 3254 3289 3292 3296 3300 3303 3330 3333 3312 3315 3318 3321 3324 3330
3336 3339 3342 3345 3348 3351 3355 3361 3368 3374 3380 3394 3398 3401 3404
3407 3410 3413 3416 3419 3422 3428 3434 3437 3440 3443 3446 3449 3453 3459
3466 3472 3478 3482 3522 3526 3530 3533 3536 3539 3545 3551 3554 3557 3560
3563 3566 3570 3576 3583 3589 3595 3609 3613 3616 3619 3625 3631 3634 3637
3640 3643 3646 3650 3654 3660 3663 3669 3675 3686 3719 3722 3726 3730 3733 3736
3742 3748 3751 3754 3757 3760 3763 3767 3773 3780 3786 3792 3806 3810 3813
3816 3822 3828 3831 3834 3837 3840 3843 3847 3853 3860 3866 3872 3880 3883 3916
3919 3923 3927 3930 3933 3936 3945 3948 3951 3954 3957 3960 3964 3970 3977
3983 3989 4003 4007 4010 4013 4019 4025 4028 4031 4034 4037 4040 4044 4050
4057 4063 4069 4080 4113 4116 4120 4124 4127 4130 4136 4142 4145 4148 4151
4154 4157 4161 4167 4174 4180 4186 4200 4204 4207 4210 4216 4222 4225 4228
4231 4234 4237 4241 4247 4254 4260 4266 4277 4310 4313 4317 4321 4324 4327
4333 4339 4342 4345 4348 4351 4354 4358 4364 4371 4377 4383 4397 4401 4404
4407 4413 4419 4422 4425 4428 4431 4434 4438 4444 4451 4457 4463 4474 4507
4510 4514 4518 4521 4524 4530 4536 4539 4542 4545 4548 4551 4555 4561 4568
4574 4580 4594 4598 4601 4604 4610 4616 4622 4625 4628 4631 4635 4641
4648 4654 4660 4671 4704 4707 4711 4715 .18 4721 4727 4733 4736 4739 4742
4745 4748 4752 4758 4765 4771 4777 4791 4798 4801 4807 4813 4816 4819
4822 4825 4828 4832 4838 4845 4851 4857 4868 4901 4904 4908 4912 4915 4918
4924 4930 4933 4936 4939 4942 4945 4949 4955 4962 4968 4974 4988 4992 4995
4998 5004 5010 5013 5016 5019 5022 5025 5029 5035 5042 5048 5054 5065 5098
5102 5106 5109 5112 5115 5121 5127 5130 5133 5137 5147 5156 5165 5176 5179
5183 5186 5191 5197 5200 5203 5207 5217 5226 5235 5243 5276 5279 5283 5288
5293 5296 5301 5307 5310 5313 5316 5319 5322 5325 5329 5335 5342 5348
5362 5366 5371 5374 5377 5380 5386 5392 5395 5398 5401 5404 5407 5410 5414
5420 5427 5433 5439 5450 5483 5486 5490 5493 5496 5499 5502 5514 5517
5523 5526 5529 5532 5536 5542 5549 5555 5561 5575 5578 5581 5584 5587
5593 5596 5602 5605 5608 5611 5614 5617 5621 5627 5634 5640 5646 5657 5690
5693 5697 5700 5703 5706 5709 5715 5721 5724 5727 5730 5733 5736 5739 5743



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	10056	10064	10067	10070	10073	10076	10079	10083	10089	10096	10102
10108	10111	10114	10120	10123	10129	10135	10138	10141	10144	10147	10150	10154	10160	10167
10173	10179	10184	10189	10192	10196	10238	10241	10244	10247	10253	10256	10259	10262	10265
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	10366	10369	10375	10381	10384	10387	10390	10393	10396	10400	10406	10413	10419	10425
10428	10431	10437	10440	10446	10449	10455	10458	10461	10464	10467	10471	10477	10484	10490
10496	10511	10546	10549	10553	10556	10559	10561	10564	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	10673	10677
10683	10686	10692	10698	10701	10704	10707	10710	10713	10717	10723	10730	10736	10742	10745
10748	10754	10757	10763	10769	10772	10775	10778	10781	10784	10788	10794	10801	10807	10813
10820	10863	10866	10869	10872	10875	10878	10881	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	10990	10994	11000
11003	11009	11015	11018	11021	11024	11027	11030	11034	11040	11047	11053	11059	11062	11065
11071	11074	11080	11086	11089	11092	11095	11098	11101	11105	11111	11118	11124	11130	11145
11180	11183	11186	11189	11192	11195	11198	11204	11207	11210	11213	11216	11219	11222	11225
11228	11231	11234	11237	11240	11243	11246	11249	11252	11255	11258	11261	11264	11267	11270
11273	11276	11279	11282	11285	11288	11291	11294	11297	11300	11303	11307	11311	11317	11320
11326	11332	11335	11338	11341	11344	11347	11361	11357	11364	11370	11376	11379	11382	11388
11391	11397	11403	11406	11409	11412	11415	11418	11421	11428	11435	11441	11447	11462	11497
11500	11503	11506	11509	11512	11515	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	11624	11628	11634	11637	11643
11649	11652	11655	11658	11661	11664	11668	11624	11681	11687	11693	11696	11699	11705	11708
11714	11720	11723	11726	11729	11732	11735	11739	11745	11752	11758	11764	11779	11814	11817
11820	11823	11826	11829	11832	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	11941	11945	11951	11954	11960	11966
11969	11972	11975	11978	11981	11985	11991	11998	12004	12010	12013	12016	12022	12025	12031
12037	12040	12043	12046	12049	12052	12056	12059	12062	12075	12081	12096	12131	12134	12137
12140	12143	12146	12149	12155	12158	12161	12164	12167	12170	12173	12176	12179	12182	12185
12188	12191	12194	12197	12210	12203	12206	12209	12212	12215	12218	12221	12224	12227	12230
12233	12236	12239	12242	12245	12246	12251	12254	12257	12260	12263	12271	12277	12283	12286
12289	12292	12295	12298	12302	12308	12315	12321	12324	12327	12330	12333	12339	12342	12348	12354
12357	12360	12363	12366	12373	12376	12379	12386	12389	12392	12395	12413	12448	12451	12454	12457
12460	12463	12466	12472	12477	12480	12481	12484	12487	12490	12493	12496	12498	12502	12505
12508	12511	12514	12517	12523	12526	12529	12532	12535	12538	12541	12544	12547	12550
12553	12556	12559	12562	12565	12568	12571	12575	12579	12582	12585	12588	12594	12600	12603	12606
12609	12612	12615	12619	12625	12628	12638	12644	12647	12650	12656	12659	12665	12671	12674
12677	12680	12683	12686	12692	12698	12703	12709	12715	12730	12765	12768	12771	12774	12777
12780	12783	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	12883	12886	12889	12892	12896	12902	12905	12911	12917	12920	12923	12926
12929	12933	12936	12942	12949	12955	12961	12964	12967	12973	12976	12982	12988	12991	12994
12997	13000	13003	13007	13013	13020	13026	13032	13047	13082	13085	13088	13091	13094	13097
13100	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	13191	13184	13187	13190
13193	13196	13199	13202	13205	13209	13213	13219	13222	13226	13234	13237	13240	13243	13246
13249	13253	13256	13260	13272	13278	13281	13284	13290	13293	13296	13305	13308	13311	13314
13317	13320	13324	13330	13337	13343	13349	13364	13399	13402	13405	13408	13411	13414	13417
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 13526 13530 13536 13539 13545 13551 13554 13557 13560 13563 13566
13570 13576 13583 13589 13595 13601 13607 13610 13616 13622 13625 13631 13634 13637 13640
13642 13651 13654 13657 13667 13670 13674 13679 13682 13686 13695 13698 13702 13706 13709
13712 13715 13718 13721 13727 13730 13733 13736 13739 13742 13745 13748 13751 13754 13757
13762 13771 13774 13780 13790 13796 14009 14021 14027 14031 14034 14046 14049 14053 14059
14063 14069 14073 14079 14083 14089 14093 14098 14103 14109 14113 14121 14125 14131 14135
14141 14145 14151 14155 14161 14165 14171 14175 14181 14185 14191 14195 14201 14207 14211
14217 14220 14223 14226 14230 14236 14239 14243 14249 14253 14256 14262 14265 14269 14275
14278 14282 14288 14291 14295 14301 14304 14308 14314 14317 14321 14327 14330 14334 14343
14346 14349 14352 14355 14361 14367 14370 14373 14376 14380 14408 14417 14420 14426 14432
14435 14438 14441 14444 14448 14481 14484 14487 14490 14493 14496 14499 14502 14505 14508
14511 14514 14517 14520 14523 14526 14529 14532 14535 14538 14541 14544 14550 14553 14556
14562 14565 14568 14574 14580 14583 14586 14595 14598 14604 14607 14613 14623 14626 14632
14638 14641 14644 14647 14650 14654 14664 14699 14702 14705 14708 14711 14714 14717 14720
14723 14726 14729 14732 14738 14747 14750 14753 14758 14784 14815 14818 14821 14824 14827
14830 14833 14836 14839 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

.SETUP 1#

.TRI 1# 2306 2386 2503 2583 2700 2780 2897 2977 3092 3104 3181 3193 3309 3324
3330 3407 3422 3428 3545 3625 3742 3822 3939 4019 4136 4216 4333 4413 4530
4610 4727 4807 4924 5004 5121 5191 5301 5386 5508 5593 5715 5800 5922 6007
6129 6214 6336 6421 6543 6628 6750 6835 6957 7042 7164 7249 7419 7456 7554
7611 7696 7818 7815 7861 8054 8176 8247 8333 8407 8483 8542 8573 8626 8702
8733 8762 8905 9107 9166 9178 9231 9412 9418 9424 9483 9486 9495 9548 9729
9736 9741 9800 9803 9812 9865 10046 10052 10058 10117 10120 10129 10162 10313 10369
10375 10434 10437 10446 10499 10680 10686 10692 10751 10754 10763 10816 10997 11079 11068
11080 11133 11314 11320 11326 11385 11388 11397 11450 11631 11637 11643 11702 11715 11714
11762 11948 11954 11960 12019 12025 12031 12084 12265 12277 12336 12348 12401 12582 12594
12653 12665 12718 12899 12911 12970 12982 13035 13216 13222 13228 13287 13290 13299 13352
13533 13546 13601 13616 13642 13780 14361 14426 14536 14541 14613 14632 14720 14729 14753
14827 14836 14842 14863 14906 14912 14933 14970 14976 14991

.UNI 1# 5159 5162 5229 5236 7830 7952 7997 8003 8006 8009 8085 8091 8094 8097
8235 8238 8395 8552 8630 8633 8814 8817 8893 8896 8780 8783 8862 8865 8901
8910 8913 8916 8982 9233 9299 9550 9616 9867 9933 10184 10250 10501 10567 10818
10884 11003 11071 11135 11201 11452 11518 11769 11835 12086 12152 12271 12339 12403 12469
12588 12656 12720 12786 12905 12973 13037 13103 13354 13420 13539 13607 13671 13789 13792
13965 13968 14003 14012 14037 14389 14398 14767 14770 14773 14809 14812 14885 14888

.SACTI 1# 217
.SAPTB 1# 2358
.SAPTH 1# 481
.SAPTY 1# 1174
.SCATC 1#
.SCMTA 1# 240
.SEOP 1# 959
.SERRO 1#
.SERRT 1#
.SPOME 1# 1662
.SRDOC 1# 1336
.SREAD 1# 1233
.SSCOP 1# 1026
.STRAP 1# 1521
.STYPE 1# 1095

```
.STYPO      1#


. 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)
```