11/70-74

11/70-74 POW FAIL

CEKBGB0

AH-7984B-MC

FICHE 1 OF 1

JAN 1980

digital

MADE IN USA

IDENTIFICATION

PRODUCT CODE: AC-7983B-MC

PRODUCT NAME: CEKBGB0 PDP-11/70-74MP System Power Fail Test

PRODUCT DATE: MAY, 1979

MAINTAINER: Diagnostic Engineering

DIAGNOSTIC ENGINEER: BILL SCHLITZKUS

CONTENTS

1.0 ABSTRACT

This diagnostic serves as a replacement to DEKBGA, the 11/70 Power Fail Test. It provides all of the coverage and features of the previous power fail test along with the added features of diagnostic support for multiprocessor systems using the KB11-CM Processor and the IIST interface.

The test is divided into two sections, Section 1 and Section 2. Section 1 completely replaces DEKBGA and basically serves as a test of CPU logic validity during a power Fail sequence. This section can be run on a standard unmodified 11/70 processor as it utilizes no multiprocessor resources. Section 2 is enabled by enabling console switch 6. This section will be executed directly after section 1. This section provides testing for proper functionality of the Interprocessor Interrupt and Sanity Timer (IIST) powerfail procedures, and memory system activity during a loss of power on the system or one of its subsystems.

This diagnostic also supports the use of the UNIBUS Exerciser (UBE) to perform simulated powerfail sequences during the execution of Section 1.

2.0 REQUIREMENTS

2.1 Equipment

This test is designed to support three types of systems.

1. Standard single processor 11/70 systems with the KB-11B/C processor.
2. Standard single processor 11/74 systems with the KB-11CM processor
3. Standard multiprocessor 11/74 systems with MKA11 memory using KB-11CM central processors connected with an IIST interprocessor interface.

At least one console terminal is required for message and error reporting on any of the above specified systems.

Optional equipment includes Unibus Exercisers for all system types and mass storage devices (RP0 5/6, massbus Exercisor) for type 3 systems.

2.2 Storage

The Power Fail test runs in 12K words of memory. The first 4K is used for stack areas and common access data. The second 4K contains the program itself. The only data elements that reside here are type and error messages, and execution flow and control variables that are used

(modified) when the processor runs in multiprocessor mode. The next 4K is used as buffer space for massbus transfers that occur during test 21.

2.3 Preliminary Procedures.

All standard 11/70-74 CPU diagnostics should first be run to insure proper operation under a secure power condition. Specifically, the IIST diagnostic and Unibus Exerciser diagnostics should be run. If massbus devices are to be used for test 21 it might be wise to insure that the devices are in proper working order. The power fail diagnostic may use all massbus drives and memory boxes accessible by the CPUs participating in this diagnostic. Therefore, it is important to switch offline to participating CPUs all data storage equipment that the operator does not want corrupted by this diagnostic. Before starting the power-fail diagnostic the operator should also be sure that the CPU power-up action switches (on the IIST front panel) are all "Run or Halt" for participating CPUs, the IIST enable switches are "online", and the IIST configuration switches are all at the same system for all participating CPUs. All participating memory boxes should be online to all participating CPUs.

The operator of this test should be familiar with the MKA11 and IIST boot/control panel.

3.0 LOADING PROCEDURE

This diagnostic can be loaded within the standard XXDP loading procedure. Note that if this diagnostic is being run as part of a chain that it must be the last element in that chain.

This diagnostic is loadable under the APT system but can only be run under APT in Uniprocessor mode with the UBE enabled.

4.0 STARTING PROCEDURE

4.1 Starting Address

The starting address for this diagnostic is 200. The restart address is 220.

4.2 Control Switch Settings

The switch settings are as follows (when set to 1):

SW15 - Halt on Error
SW14 - Loop on test (Section 1 only)
SW08 - Enable System Power Fail Test (Test 25)
SW07 - Disable Section 1 Tests (Multiprocessor mode only)

SW6	SW5	
0	0	Uniprocessor mode, manual powerfail. (this mode can be used a a replacement for previous powerfail diagnostic)
0	1	Uniprocessor mode, automatic (UBE) powerfail (uniprocessor mode that should be used under APT)
1	0	Multiprocessor mode, manual powerfail. Both sections (1 and 2) are executed.
1	1	Multiprocessor mode, automatic (UBE) powerfail. Only Section 1 will use the UBE. Each CPU must have a UBE.

Note that for the multiprocessor modes (SW6 enabled) the IIST "system" ID of each CPU participating in the test should be specified by setting the appropriate bit in Switches 0-3 in the switch register. For example, if all processors in a 4 processor system are to be tested then switches 0-3 should be enabled. If only CPU's 0 and 2 are to participate then only switches 0 and 2 should be enabled.

The switches are defined by the master CPU only (in multiprocessor mode), where the master is the CPU that starts or restarts the program. Each slave CPU uses the switch definitions of the master.

The Program will not interpret changes to the switches in multiprocessor mode while the program is running. The program must be restarted.

4.3 Restarting Procedures

The restart address for this diagnostic is 220. A restart should be considered a completely new session therefore be sure the proper switches are enabled for the features that you want.

4.4 Program And/Or Operator Action

NOTE

Be sure to isolate the CPU's that are to participate in the powerfail test from other CPU's that are to remain functional. Participating CPU's should have the IIST configuration switches found on the IIST panel switched to the SYSTEM B position (if this is the default position for the installation then the SYSTEM A position could be used).

5.0 OPERATING PROCEDURES

5.1 Modes Of Operation

This d agnostic is designed to run in four basic modes specified by the selection of bits 5 & 6 in the console switch register. It should be noted that if a multiprocessor mode is chosen (SW6 enabled) that the first CPU started will become the "master" and successively start the remaining participating CPU's, the "Slaves".

5.2 Common Procedures

Load address 200 and then enable the switches for the test mode that is desired. Remember to enable the appropriate CPU mask bit if you are running a multiprocessor mode. When this has been done insure that all slaves are powered up and have their halt switches in the "enable" state if MP mode is to be used. Hit start - the program name will be typed followed by the mode the diagnostic is running in (uniprocessor or multi-processor) followed by the contents of the switch register and whether the Unibus Exerciser will be used to simulate power fails.

5.3 Notes And Warnings

1. Power failures in section 1 are only allowed when expected. Therefore in manual mode do not remove the power until the test number appears in the display register.

2. Power failures are not allowed during the execution of the End of Pass (EOP) routines. Note that the number displayed during the End of Pass is the Pass number NOT the test number.

3. When running the diagnostic in uniprocessor mode on a single CPU that is part of a multiprocessor system make sure that the IIST configuration switch for the respecitve CPU is in the STAND ALONE position and the IIST ENABLE switch is in the

OFF LINE position.

4. When running in multiprocessor mode insure that the subsystem under test is isolated from any system that is still running by switching the CPU's under test to an alternate postion with the IIST CONFIGURATION swtiches. For example, if the default system normally runs on the "SYSTEM A" postion, switch the subsystem under test to the "SYSTEM B" position.

5.4 Uniprocessor Mode, No UBE

The diagnostic will instruct you to "interrupt the power after the test number appears on the display". Interrupt the power only at this time. If the test is successful then the next test number should appear in the display. When the End of Pass is reached the Pass count will be typed. There are no error reports in Section 1 (except for unexpected traps to 4 and 114.) Normally, an error results in a processor halt.

5.5 Uniprocessor Mode, With UBE

In this mode the UBE is used to perform the power Fail procedure. No "interrupt the power..." message is typed before the UBE takes action. The test number and pass number however do appear in the display register. An EOP message is typed equivalent to that of Section 5.4. Note that in this mode only the CPU logic involved in the power-fail sequence is tested. Failures that may occur because of problems in the Power system will go undetected.

5.6 Multiprocessor Mode, No UBE

In multiprocessor mode the CPU that is initially started becomes the "MASTER" CPU. What this means is that all error messages and typeouts will appear on this CPU's console. This CPU is also responsible for startup of "slave" CPU's. Slave CPU's are the remaining CPU's that are scheduled to participate in the test by their appropriate bit being set in switches 0-3 of the master CPU's SWR. When start is depressed and multi-processor mode is specified the program name, test mode, and switch register setting will be typed upon the master CPU console. The remaining CPU's will then be booted and then interrupted through the masters IIST. This will be followed by an "interrupt the power..." message.

The ID of the processor responsible for typeouts and error messages will precede the message:

0>

n>

Run each processor through section 1 one at a time. After power-failing the last test in Section 1 (Test 16) on one CPU go on to the next CPU, etc. After the last test on the last CPU has been completed in Section 1, the operator will receive instructions at console.

Whereas there are no prompts printed at the console in Section 1, all power fails in Section 2 must be done in exact agreement with the typed-out instructions.

Section 2 prompts the operator to remove the power from a particular element of the system. If the expected results occur then the next element is tried. When all relevant elements have been tried then the diagnostic precedes to the next test. Section 2 contains tests 17 through 25.

5.7 Multiprocessor Mode, With UBE

This mode is the same as "multiprocessor mode, no UBE" except that Section 1 will be done without manual intervention. ALL CPUs that are to participate must have a UBE module.

6.0 ERRORS AND ERROR REPORTING

Error reports are always typed out on the console of the CPU that was first started. This is the "master" CPU in multiprocessor mode. All error messages are preceded by a tag as follows 'n>' where n's the IIST self-ID of the CPU encountering the error. Typing in multiprocessor mode is always done by the master. If the master is without power when typing is required, the messages will be queued and printed when power has been restored.

7.0 TEST DESCRIPTIONS

The tests that are found in section I are simple power fail tests that guarantee that the proper machine states are entered on power fail and power up. The test names are self- explanatory.

Section 1

1. Simple Down/Up test (Kernal node)
2. Program Voltility Test
   Ver'fy that the memory bank containing the program will not be corrupted by CPU power fails.
3. Simple Down/Up test (Supervisor mode)
4. Simple Down/Up Test (User mode)
5. Power fail with odd address
6. Power Fail in the Red Zone
7. Power Fail with memory timeout (kernai)
10. Power Fail in the yellow zone.
11. Power Fail with resets.
12. Power Fail with odd address (Supervisor).
13. Power Fail with Timeout (Supervisor)
14. Power Fail with odd address (User)
15. Power Fail with timeout (User)
16. Memory Management Abort Test

Section 2

After a'l CPUs reach the beginning of Section 2, each CPU sizes for RP04/5/6 massbus devices. If no devices are found the following message is printed:

No Massbus Device Available On CPU #n

If the only massbus device found has its PGM bit set, the following message is printed:

Warning: Drive #n
On CPU #n
Is accessible over Ports A and B

and will be used later in this diagnostic.

17. Check 'BRK' & 'DCF' FLAGS during power fail. Insure that the IIST's of functioning CPU's receive BRK & DCF signals corresponding to the CPU that performed a Power Fail.

The operator will be prompted with messages of the following type:

Power Fail CPU #n

Afte restoring power, each of the other CPUs should report:

CPU Interrupt As Expected

20. Check power fail during high memory activity. Insure that power down sequences can be performed by a CPU while other CPU's are contending for the memory bus.

The operator will be prompted with messages of the following type:

Power Fail CPU #n

There is no report from the other CPUs after restoring power.

21. Check power fail during massbus transfer. Insure that power down sequences can be performed by a CPU while other CPU's are performing massbus read operations. Also verify that the read operations don't experience any loss of data due to the power condition of an uninvolved CPU.

The operator will be prompted with messages of the following type:

Power Fail CPU #n

If there are no massbus devices for the other CPUs to use, then the following messages are printed instead of the above message:

No Massbus Device Available On CPU #n Proceeding
To Next CPU.

There is no report from the other CPUs after restoring power.

22. Insure that a loss of AC power on a MKA11 semiconductor memory box does not cause a power fail sequence to occur on any processor that has a disabled part to that box.

    The operator will be prompted with messages of the type:

        Get Set To Power Fail Mem Box #n

    Put battery backup on all memory boxes
    Make all memory ports offline only on mem box to be power failed
    Make all CPU power-up switches "Run or Halt"

    Now Power Fail The Mem Box

    Restore power 5 seconds after power fail
    Restore all memory ports online
    Then type any character at the master console
    No CPU should report a power fail

    The master should report "OK".

23. Check AC power fail on memory box. Insure that a loss of AC power on a MKA11 semiconductor memory box causes a power fail sequence to occur on any processor that has an enabled port to that box.

    The operator will be prompted with message of the type:

        Get Set To Power Fail Mem Box #n

    Put battery backup on all memory boxes
    Make all memory ports online
    Make all CPU power-up switches "Run or Halt"

    Now power fail the mem box

    Restore power 5 seconds after power fail then type any character at the master console. Each CPU should report a power failure.

    Each CPU should report the following message:

        Power Failure On CPU As Expected

24. Check DC Power Loss on a memory box. Insure that the slave CPU(s) specified through the IIST Boot/control panel perform a boot operation when AC & DC power are restored to the memory box.

    The operator will be prompted with message of the type:

        Get Set To Power Fail Mem Box #n

Disable battery backup on mem box to be power-failed. Put all slave CPU mem ports online. Make master CPU mem port offline only on box to be power-failed. Make all CPU power-up switches 'RUN OR Boot'.

Now Power Fail The Mem Box

Restore power 5 seconds after power fail
Restore all mem ports online
Restore all CPU power-up switches to 'Run or Halt''
Then type any character at the master console
Each slave should report an interrupt

Each slave CPU should report the following message:

CPU Interrupt As Expected

25. Check system recovery on power fail. Insure that a total momentary loss of AC power on a system level is recoverable without operator intervention.

The operator will be prompted with the following message:

Get Set to Power Fail Entire System...

Put battery backup on all mem boxes
Make all memory ports online
Make all CPU power-up switches 'Run or Halt''

Now Power Fail the Entire System
Restore power 5 seconds after power fail

Each CPU should report the following:

8.0 MISCELLANEOUS

Test 24 will not be done on the memory box with base address 0. Therefore, in order to fully test all system elements the operator should restart this diagnostic a second time and switch the box with base address 0 with another box (using the thumbwheels to switch base addresses). The operator should also switch master CPUs on the restart.

SEQ 0013

Test 22 will be skipped if there is only one MKA11 memory box. Power Failing through Section 1 with the UBE will not necessarily test power fail during odd address trap, timeout, etc. (it depends on when the UBE starts the power down). Therefore, it is recommended to manually power fail Section 1.

```
.TITLE  MAINDEC-11-CEKBG-B      PDP-11/70,74 SYSTEM  POWER FAIL
;*COPYRIGHT (C) 1978
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;*
;*PROGRAM BY JIM LACEY, JEFF WHITE, BILL SCHLITZKUS
;*
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
;*

;;*******************************************************************
;;*                                                                 *
;;*              11-70/74 SYSTEM POWER FAIL DIAGNOSTIC              *
;;*              -------------------------------------              *
;;*                                                                 *
;;*  THIS DIAGNOSTIC IS DIVIDED INTO TWO SECTIONS: SECTION 1 TESTS THE *
;;* BASIC ABILITY OF A PROCESSOR TO SUCCESSFULLY ENTER AND RECOVER  *
;;* FROM A POWER FAIL CONDITION.  THIS SECTION REPLACES, IN FUNC-   *
;;* TIONALITY, THE PREVIOUS POWER FAIL DIAGNOSTIC DEKBGA AND PROVIDES *
;;* EQUIVALENT DIAGNOSTIC COVERAGE.  THIS SECTION AND ONLY THIS     *
;;* SECTION  WILL BE RUN IF THE MP SWITCH (SWITCH 6) IS DISABLED.   *
;;*       SECTION 2 OF THIS DIAGNOSTIC PROVIDES DIAGNOSTIC COVERAGE *
;;* FOR MULTIPROCESSOR CONFIGURATIONS UTILIZING THE IIST INTERFACE  *
;;* AS A MEANS OF INTERPROCESSOR COMMUNICATION.  IF THE MP SWITCH   *
;;* (SWITCH 6) IS ENABLED BOTH SECTION 1 AND SECTION 2 ARE PERFORMED, *
;;* ALSO THE IIST IS USED IN SECTION 1 TO INITIALIZE AND START ALL  *
;;* PARTICIPATING PROCESSORS.  SECTION 2 TESTS THE ABILITY OF A MULTI- *
;;* PROCESSOR SYSTEM TO SUCCESSFULLY RECOVER FROM A POWER FAILURE   *
;;* EITHER IN A SELECTIVE SUBSYSTEM (MEMORY BOX OR PROCESSOR) OR ON *
;;* A SYSTEM WIDE LEVEL DURING VARIOUS KINDS OF MEMORY AND I/O ACTIV- *
;;* ITY.                                                            *
;;*  IN THE MULTIPROCESSOR MODE, ALL CPUS MUST ARRIVE AT THE        *
;;* ENTRY POINT TO SECTION 2 BEFORE ANY CPU WILL BEGIN SECTION 2    *
;;* TESTING.  IF SECTION 1 IS NOT SKIPPED AND THE UNIBUS EXCERCISER *
;;* IS NOT BEING USED, THE OPERATOR MUST POWER FAIL EACH CPU        *
```

```
                        ;;* MANUALLY THRU THE 16 TESTS OF SECTION 1 TO GET THE CPU TO THE      *
                        ;;* ENTRY POINT OF SECTION 2.                                          *
                        ;;*     BEFORE STARTING THE PROGRAM, BE SURE THAT THE CPU POWER-UP     *
                        ;;* ACTION SWITCHES (ON THE IIST FRONT PANEL) ARE ALL                  *
                        ;;* 'RUN OF HALT'', THE IIST ENABLE SWITCHES ARE ONLINE, AND           *
                        ;;* THE IIST CONFIGURATION SWITCHES ARE EITHER ALL SYSTEM 0 OR 1.      *
                        ;;* ALL PARTICIPATING MEMORY BOXES SHOULD BE ONLINE TO ALL PARTIC-     *
                        ;;* IPATING CPUS.                                                      *
                        ;;*********************************************************************

                        ;SWITCH REGISTFR DEFINITIONS
                        ;
                        ;THE SWITCHES ARE DEFINED BY THE MASTER CPU ONLY,
                        ;WHERE THE MASTER IS THE CPU THAT STARTS OR RESTARTS
                        ;THE PROGRAM.  EACH SLAVE CPU USES THE SWITCH DEFINITIONS
                        ;OF THE MASTER.
                        ;
                        ;THE PROGRAM WILL NOT INTERPRET CHANGES TO THE
                        ;SWITCHES IN MULTIPROCESSOR MODE WHILE THE PROGRAM IS RUNNING.  THE
                        ;PROGRAM MUST BE RESTARTED.
                        ;
                        ;SW15=1          HALT ON ERROR
                        ;SW14=1          LOOP ON TEST (SECTION 1 ONLY)
                        ;SW08=1          ENABLE SYSTEM POWER FAIL TEST (TEST 25)
                        ;SW07=1          DISABLE SECTION 1 TESTS (MULTIPROCESSOR MODE ONLY)
                        ;SW06=1          ENABLE MULTIPROCESSOR MODE/SECTION 2 TESTS
                        ;SW05=1          ENABLE UNIBUS EXERCISERS (SECTION 1 ONLY, EACH CPU MUST HAVE A UBE)
                        ;SW03=1          TEST CPU #3 (IIST SELF ID), MULTIPROCESSOR MODE ONLY
                        ;SW02=1          TEST CPU #2        ''                    ''
                        ;SW01=1          TEST CPU #1        ''                    ''
                        ;SW00=1          TEST CPU #0        ''                    ''
                        .SBTTL  BASIC DEFINITIONS

                        ;*INITIAL ADDRESS OF THE STACK POINTER
013776                  STACK=  13776           ;;FIRST ADDRESS OF THE STACK
013776                  KERSTK= STACK           ;;KERNEL STACK
013576                  SUPSTK= STACK-200       ;;SUPERVISOR STACK
013476                  USESTK= STACK-300       ;;USER STACK
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
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
177546                  LKS=    177546          ;;LINE CLOCK (KW11-L) STATUS REGISTER

                        ;*MISCELLANEOUS DEFINITIONS
000011                  HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                  LF=     12              ;;CODE LINE FEED
000015                  CR=     15              ;;CODE CARRIAGE RETURN
000200                  CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
```

```
                        R1=     %1              ;;GENERAL REGISTER
                        R2=     %2              ;;GENERAL REGISTER
                        R3=     %3              ;;GENERAL REGISTER
                        R4=     %4              ;;GENERAL REGISTER
                        R5=     %5              ;;GENERAL REGISTER
                        R6=     %6              ;;GENERAL REGISTER
                        R7=     %7              ;;GENERAL REGISTER
                        .EQUIV  R0,R10          ;;GENERAL REGISTER
                        .EQUIV  R1,R11          ;;GENERAL REGISTER
                        .EQUIV  R2,R12          ;;GENERAL REGISTER
                        .EQUIV  R3,R13          ;;GENERAL REGISTER
                        .EQUIV  R4,R14          ;;GENERAL REGISTER
                        .EQUIV  R5,R15          ;;GENERAL REGISTER
                        SP=     %6              ;;STACK POINTER
                        .EQUIV  SP,KSP          ;;KERNEL STACK POINTER
                        .EQUIV  SP,SSP          ;;SUPERVISOR STACK POINTER
                        .EQUIV  SP,USP          ;;USER STACK POINTER
                        PC=     %7              ;;PROGRAM COUNTER

                        ;*PRIORITY LEVE_ DEFINITIONS
                        PR0=    0               ;;PRIORITY LEVEL 0
                        PR1=    40              ;;PRIORITY LEVEL 1
                        PR2=    100             ;;PRIORITY LEVEL 2
                        PR3=    140             ;;PRIORITY LEVEL 3
                        PR4=    200             ;;PRIORITY LEVEL 4
                        PR5=    240             ;;PRIORITY LEVEL 5
                        PR6=    300             ;;PRIORITY LEVEL 6
                        PR7=    340             ;;PRIORITY LEVEL 7

                        ;*''SWITCH REGISTER'' SWITCH DEFINITIONS
                        SW15=   100000
                        SW14=   40000
                        SW13=   20000
                        SW12=   10000
                        SW11=   4000
                        SW10=   2000
                        SW09=   1000
                        SW08=   400
                        SW07=   200
                        SW06=   100
                        SW05=   40
                        SW04=   20
                        SW03=   10
                        SW02=   4
                        SW01=   2
                        SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0
```

```
                                ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                          BIT15=  100000
040000                          BIT14=  40000
020000                          BIT13=  20000
010000                          BIT12=  10000
004000                          BIT11=  4000
002000                          BIT10=  2000
001000                          BIT09=  1000
000400                          BIT08=  400
000200                          BIT07=  200
000100                          BIT06=  100
000040                          BIT05=  40
000020                          BIT04=  20
000010                          BIT03=  10
000004                          BIT02=  4
000002                          BIT01=  2
000001                          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
000100                          LKVEC=  100             ;;LINE CLOCK (KW11-L) VECTER
000114                          CACHVEC=114             ;;CACHE ERROR INTERRUPT VECTOR
000240                          PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
000250                          MMVEC=  250             ;;MEMORY MANAGEMENT VECTOR
                                .SBTTL  CACHE   REGISTER DEFINITIONS


177740                          LOADRS = 177740         ;;LOWER 16 BITS OF ADDRESS THAT CAUSED ERROR
177742                          HIADRS = 177742         ;;UPPER SIX BITS OF ADDRESS THAT CAUSED ERROR
177744                          MEMERR = 177744         ;;CACHE ERROR REGISTER
177746                          CONTRL = 177746         ;;MEMORY CONTROL REGISTER
177750                          MAINT  = 177750         ;;MEMORY MAINTENENCE REGISTER
177752                          HITMIS = 177752         ;;HIT MISS REGISTER "1" IMPLIES HIT IN CACHE

                                .SBTTL  CPU REGISTER DEFINITIONS
```

```
                                ;;MEMORY SIZE REGISTER NUMBER TO PUT INTO A PAR
                                ;;TO GET TO THE LAST 32 WORDS OF MEMORY
                                ;;HIGH SIZE REGISTER, RESERVED FOR FUTURE USE
                                ;;CURRENTLY ALL ZERO
                                ;;SYSTEM ID REGISTER
                                ;;CPU ERROR REGISTER HOLDS CONDITION THAT CAUSED
                                ;;THE TRAP TO ERRVEC (000004)


                                .SBTTL  MEMORY MANAGEMENT DEFINITIONS


                                ;*MEMORY MANAGEMENT STATUS REGISTER ADDRESSES

177572                          MMR0=   177572
177574                          MMR1=   177574
177576                          MMR2=   177576
172516                          MMR3=   172516
                                .EQUIV  MMR0,SR0
                                .EQUIV  MMR1,SR1
                                .EQUIV  MMR2,SR2
                                .EQUIV  MMR3,SR3

                                ;*USER "I" PAGE DESCRIPTOR REGISTERS

177600                          UIPDR0= 177600
177602                          UIPDR1= 177602
177604                          UIPDR2= 177604
177606                          UIPDR3= 177606
177610                          UIPDR4= 177610
177612                          UIPDR5= 177612
177614                          UIPDR6= 177614
177616                          UIPDR7= 177616

                                ;*USER "D" PAGE DESCRIPTOR REGISTORS

177620                          UDPDR0= 177620
177622                          UDPDR1= 177622
177624                          UDPDR2= 177624
177626                          UDPDR3= 177626
177630                          UDPDR4= 177630
177632                          UDPDR5= 177632
177634                          UDPDR6= 177634
177636                          UDPDR7= 177636

                                ;*USER "I" PAGE ADDRESS REGISTERS

177640                          UIPAR0= 177640
177642                          UIPAR1= 177642
177644                          UIPAR2= 177644
177646                          UIPAR3= 177646
177650                          UIPAR4= 177650
177652                          UIPAR5= 177652
```

```
                177654          UIPAR6= 177654
                177656          UIPAR7= 177656

                                ;*USER 'D' PAGE ADDRESS REGISTERS

                177660          UDPAR0= 177660
                177662          UDPAR1= 177662
                177664          UDPAR2= 177664
                177666          UDPAR3= 177666
                177670          UDPAR4= 177670
                177672          UDPAR5= 177672
                177674          UDPAR6= 177674
                177676          UDPAR7= 177676

                                ;*SUPERVISOR 'I' PAGE DESCRIPTOR REGISTERS

                172200          SIPDR0= 172200
                172202          SIPDR1= 172202
                172204          SIPDR2= 172204
                172206          SIPDR3= 172206
                172210          SIPDR4= 172210
                172212          SIPDR5= 172212
                172214          SIPDR6= 172214
                172216          SIPDR7= 172216

                                ;*SUPERVISOR 'D' PAGE DESCRIPTOR REGISTERS

                172220          SDPDR0= 172220
                172222          SDPDR1= 172222
                172224          SDPDR2= 172224
                172226          SDPDR3= 172226
                172230          SDPDR4= 172230
                172232          SDPDR5= 172232
                172234          SDPDR6= 172234
                172236          SDPDR7= 172236

                                ;*SUPERVISOR 'I' PAGE ADDRESS REGISTERS

                172240          SIPAR0= 172240
                172242          SIPAR1= 172242
                172244          SIPAR2= 172244
                172246          SIPAR3= 172246
                172250          SIPAR4= 172250
                172252          SIPAR5= 172252
                172254          SIPAR6= 172254
                172256          SIPAR7= 172256

                                ;*SUPERVISOR 'D' PAGE ADDRESS REGISTERS

                172260          SDPAR0= 172260
                172262          SDPAR1= 172262
                172264          SDPAR2= 172264
                172266          SDPAR3= 172266
                172270          SDPAR4= 172270
                172272          SDPAR5= 172272
                172274          SDPAR6= 172274
```

```
                        ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS

172300                  KIPDR0= 172300
172302                  KIPDR1= 172302
172304                  KIPDR2= 172304
172306                  KIPDR3= 172306
172310                  KIPDR4= 172310
172312                  KIPDR5= 172312
172314                  KIPDR6= 172314
172316                  KIPDR7= 172316

                        ;*KERNEL "D" PAGE DESCRIPTOR REGISTERS

172320                  KDPDR0= 172320
172322                  KDPDR1= 172322
172324                  KDPDR2= 172324
172326                  KDPDR3= 172326
172330                  KDPDR4= 172330
172332                  KDPDR5= 172332
172334                  KDPDR6= 172334
172336                  KDPDR7= 172336

                        ;*KERNEL "I" PAGE ADDRESS REGISTERS

172340                  KIPAR0= 172340
172342                  KIPAR1= 172342
172344                  KIPAR2= 172344
172346                  KIPAR3= 172346
172350                  KIPAR4= 172350
172352                  KIPAR5= 172352
172354                  KIPAR6= 172354
172356                  KIPAR7= 172356

                        ;*KERNEL "D" PAGE ADDRESS REGISTERS

172360                  KDPAR0= 172360
172362                  KDPAR1= 172362
172364                  KDPAR2= 172364
172366                  KDPAR3= 172366
172370                  KDPAR4= 172370
172372                  KDPAR5= 172372
172374                  KDPAR6= 172374
172376                  KDPAR7= 172376


                        .SBTTL  UNIBUS MAP REGISTER DEFINITIONS


                        ;*THE LOWER 16 BITS OF THE MAP REGISTERS ARE LABELED 'MAPLXX'
                        ;*THE UPPER 6 BITS OF THE MAP REGISTERS ARE LABELED 'MAPHXX'


170200                  MAPL00 = 170200
```

```
393     170202          MAPH00 = 170202
394     170204          MAPL01 = 170204
395     170206          MAPH01 = 170206
396     170210          MAPL02 = 170210
397     170212          MAPH02 = 170212
398     170214          MAPL03 = 170214
399     170216          MAPH03 = 170216
400     170220          MAPL04 = 170220
401     170222          MAPH04 = 170222
402     170224          MAPL05 = 170224
403     170226          MAPH05 = 170226
404     170230          MAPL06 = 170230
405     170232          MAPH06 = 170232
406     170234          MAPL07 = 170234
407     170236          MAPH07 = 170236
408     170240          MAPL10 = 170240
409     170242          MAPH10 = 170242
410     170244          MAPL11 = 170244
411     170246          MAPH11 = 170246
412     170250          MAPL12 = 170250
413     170252          MAPH12 = 170252
414     170254          MAPL13 = 170254
415     170256          MAPH13 = 170256
416     170260          MAPL14 = 170260
417     170262          MAPH14 = 170262
418     170264          MAPL15 = 170264
419     170266          MAPH15 = 170266
420     170270          MAPL16 = 170270
421     170272          MAPH16 = 170272
422     170274          MAPL17 = 170274
423     170276          MAPH17 = 170276
424     170300          MAPL20 = 170300
425     170302          MAPH20 = 170302
426     170304          MAPL21 = 170304
427     170306          MAPH21 = 170306
428     170310          MAPL22 = 170310
429     170312          MAPH22 = 170312
430     170314          MAPL23 = 170314
431     170316          MAPH23 = 170316
432     170320          MAPL24 = 170320
433     170320          MAPH24 = 170320
434     170324          MAPL25 = 170324
435     170326          MAPH25 = 170326
436     170330          MAPL26 = 170330
437     170332          MAPH26 = 170332
438     170334          MAPL27 = 170334
439     170336          MAPH27 = 170336
440     170340          MAPL30 = 170340
441     170342          MAPH30 = 170342
442     170344          MAPL31 = 170344
443     170346          MAPH31 = 170346
444     170350          MAPL32 = 170350
445     170352          MAPH32 = 170352
446     170354          MAPL33 = 170354
447     170356          MAPH33 = 170356
448     170360          MAPL34 = 170360
```

```
                        MAPH34 = 170362
                        MAPL35 = 170364
                        MAPH35 = 170366
                        MAPL36 = 170370
                        MAPH36 = 170372
                        MAPL37 = 170374
                        MAPH37 = 170376
                        .EQUIV  MAPL00,MAPL0
                        .EQUIV  MAPH00,MAPH0
                        .EQUIV  MAPL01,MAPL1
                        .EQUIV  MAPH01,MAPH1
                        .EQUIV  MAPL02,MAPL2
                        .EQUIV  MAPH02,MAPH2
                        .EQUIV  MAPL03,MAPL3
                        .EQUIV  MAPH03,MAPH3
                        .EQUIV  MAPL04,MAPL4
                        .EQUIV  MAPH04,MAPH4
                        .EQUIV  MAPL05,MAPL5
                        .EQUIV  MAPH05,MAPH5
                        .EQUIV  MAPL06,MAPL6
                        .EQUIV  MAPH06,MAPH6
                        .EQUIV  MAPL07,MAPL7
                        .EQUIV  MAPH07,MAPH7
170016                          UBCR2=170016



                        .SBTTL  IIST REGISTER DEFINITIONS


                        ;; IIST INTERNAL REGISTERS

000000                          PGTE = 0        ;;PROGRAM-GENERATED TRANSMISSION ENABLE
000001                          PGCS = 1        ;;PROGRAM-GENERATED CONTROL STATUS
000002                          STTE = 2        ;;SANITY-TIMER TRANSMISSION ENABLE
000003                          STCS = 3        ;;SANITY-TIMER CONTROL STATUS
000004                          IMSK = 4        ;;INPUT MASK
000005                          PGF = 5         ;;PROGRAM GENERATED FLAGS
000006                          STF = 6         ;;SANITY-TIMER FLAGS
000007                          DCF = 7         ;;DCLO/DISCONNECT FLAGS
000010                          EXC = 10        ;;EXCEPTIONS
000015                          MTC = 15        ;;MAINTAINANCE CONTROL

                        ;;IIST INTERRUPT VECTOR


                        .SBTTL  CONSOLE SWITCH SETTINGS
                                ;; WHEN THESE SWITCHES ARE ENABLED, IT SPECIFIES TO THE MASTER
                                ;; THAT THE CPU WITH THE IIST SELF ID CORRESPONDING TO THE
                                ;;EQUIVILENT BIT POSITION IS EXPECTED TO PARTICIPATE IN THIS TEST

000001                          CP0=BIT0                        ;CPU0 MASK LOCATION
000002                          CP1=BIT1                        ;CPU1 MASK LOCATION
000004                          CP2=BIT2                        ;CPU2 MASK LOCATION
000010                          CP3=BIT3                        ;CPU3 MASK LOCATION
```

```
                                        ;; THE FOLLOWING SWITCHES CONTROL THE EXECUTION STREAM OF THE
                                        ;; TEST.
                                        
000100                                  MPSW= BIT6              ;ENABLE FOR MULTIPROCESSOR CONFIGURATIONS
000040                                  UBESW= BIT5             ;ENABLE IF UNIBUS EXERCISER IS TO BE USED
001000                                  LOE= BIT9               ;ENABLE FOR LOOPING ON ERRORS
040000                                  LOT= BIT14              ;ENABLE FOR LOOPING ON TEST
100000                                  HOE= BIT15              ;HALT ON ERROR
```

```
                                .SBTTL  RJP04 DEVICE REGISTERS
176700                          RPCS1=176700            ;;CONTROL AND STATUS 1
176702                          RPWC =176702            ;;WORD COUNT REGISTER
176704                          RPBA =176704            ;;UNIBUS ADDRESS
176706                          RPDA =176706            ;;DESIRED SECTOR/TRACK ADDRESS
176710                          RPCS2=176710            ;;CONTROL AND STATUS 2
176712                          RPDS =176712            ;;DRIVE STATUS
176714                          RPER1=176714            ;;ERROR REGISTER 1
176716                          RPAS =176716            ;;ATTENSION SUMMARY
176720                          RPLA =176720            ;;LOOK-AHEAD REGISTER
176722                          RPDB =176722            ;;DATA BUFFER REGISTER
176724                          RPMR =176724            ;;MAINTENANCE REGISTER
176726                          RPDT =176726            ;;DRIVE TYPE
176730                          RPSN =176730            ;;SERIAL NUMBER REGISTER
176732                          RPOF =176732            ;;OFFSET REGISTER
176734                          RPDC =176734            ;;DESIRED CYLINDER REGISTER
176736                          RPCC =176736            ;;CURRENT CYLINDER REGISTER
176740                          RPER2-176740            ;;ERROR REGISTER 2
176742                          RPER3=176742            ;;ERROR REGISTER 3
176744                          RPEC1-176744            ;;ECC POSITION REGISTER
176746                          RPEC2-176746            ;;ECC PATTERN REGISTER
```

```
                                .SBTTL  POWER FAIL FUNCTION TABLE BIT DIFINITIONS

            010000                      NCX=BIT12               ;;DON,T SAVE MM REGISTERS
            004000                      TI =BIT11               ;;TIME THE POWER FAIL
            002000                      NS =BIT10               ;;DON'T PERFORM A REGISTER SAVE
            001000                      SID=BIT9                ;;SEND ERROR ON ILLEGAL DOWN
            000400                      SIU=BIT8                ;;SEND ERROR ON ILLEGAL UP
            000200                      SED=BIT7                ;;SEND ERROR ON DOWN
            000100                      SEU=BIT6                ;;SED ERROR ON UP
            000040                      SSD=BIT5                ;;SEND SIGNAL ON DOWN
            000020                      SSU=BIT4                ;;SEND SIGNAL ON UP.
                                .SBTTL  TRAP CATCHER

            000000                      .=0
                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
            000174                      .=174
000174  000000                 DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000                 SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                                .SBTTL  ACT11 HOOKS

                                ;;**************************************************************************
                                ;HOOKS REQUIRED BY ACT11
            000200                      $SVPC=.                 ;SAVE PC
            000046                      .=46
000046  035644                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
            000052                      .=52
000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
            000200                      . $SVPC                 ;; RESTORE PC
```

```
                                                .SBTTL  LOAD START AND RESTART VECTORS
                    000200                      .=200
000200  000137  020064                          JMP     STRT            ;LOAD 200 WITH A JUMP TO START OF TEST
                    000220                      .=220
000220  004737  020000                          JSR     PC,     RESTRT          ;LOAD 220 WITH A JUMP TO THE RESTART CODE
000224  000137  020064                          JMP     STRT

```

```
                                .SBTTL  APT PARAMETER BLOCK

                                ;;**********************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;**********************************************************************
000230                                  .$X=.   ;;SAVE CURRENT LOCATION
000024                                  .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                          200     ;;FOR APT START UP
000044                                  .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  000230                          $APTHDR ;;POINT TO APT HEADER BLOCK
000230                                  .=.$X   ;;RESET LOCATION COUNTER
                                ;;**********************************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                ;INTERFACE SPEC.

000230                          $APTHD:
000230  000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
000232  014336                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
000234  000000                  $TSTM:  .WORD           ;;RUN TIM OF LONGEST TEST
000236  000000                  $PASTM: .WORD           ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
000240  000000                  $UNITM: .WORD           ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
000242  000052                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                                .SBTTL  COMMON TAGS

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

                        014000          .=14000
014000                  $CMTAG:                         ;;START OF COMMON TAGS
014000  000000                  .WORD   0
014002      000         $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
014003      000                 .BYTE   0
014004      000                 .BYTE   0
014005      000                 .BYTE   0
014006      000         $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
014007      000                 .BYTE   0
014010      000                 .BYTE   0
014011      000                 .BYTE   0
014012  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
014014  000000                  .WORD   0
014016  000000                  .WORD   0
014020  000000                  .WORD   0
014022  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
014024  000000                  .WORD   0
014026  000000                  .WORD   0
014030  000000                  .WORD   0
014032  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
014034  000000                  .WORD   0
014036  000000                  .WORD   0
014040  000000                  .WORD   0
014042  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
014044  000000                  .WORD   0
014046  000000                  .WORD   0
014050  000000                  .WORD   0
014052  000000                  .WORD   0
014054      000         $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
014055      000                 .BYTE   0
014056      000                 .BYTE   0
014057      000                 .BYTE   0
014060      001         $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
014061      001                 .BYTE   1
014062      001                 .BYTE   1
014063      001                 .BYTE   1
014064  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
014066  000000                  .WORD   0
014070  000000                  .WORD   0
014072  000000                  .WORD   0
014074  000000          $ERRSP: .WORD   0               ;;CONTAINS SP OF CPU IN ERROR
014076  000000                  .WORD   0
014100  000000                  .WORD   0
014102  000000                  .WORD   0
014104  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
014106  000000                  .WORD   0
014110  000000                  .WORD   0
014112  000000                  .WORD   0
014114  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
014116  000000                  .WORD   0
```

```
                                                .WORD   0
                                                .WORD   0
                                $GDDAT:         .WORD   0               ;;CONTAINS 'GOOD' DATA
                                                .WORD   0
                                                .WORD   0
                                                .WORD   0
                                $EOPSG:         .BYTE   0               ;;THIS TABLE HOLDS THE END OF PASS
                                                .BYTE   0
                                                .BYTE   0
                                                .BYTE   0

                                $BDDAT:         .WORD   0               ;;CONTAINS 'BAD' DATA
                                                .WORD   0
                                                .WORD   0
                                                .WORD   0
                                                .WORD   0               ;;RESERVED--NOT TO BE USED
                                                .WORD   0
                                $AUTOB:         .BYTE   0               ;;AUTOMATIC MODE INDICATOR
                                $INTAG:         .BYTE   0               ;;INTERRUPT MODE INDICATOR
                                                .WORD   0
                                SWR:            .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
                                                .WORD   DSWR
                                                .WORD   DSWR
                                                .WORD   DSWR
                                DISPLAY:        .WORD   DDISP           ;;ADDRESS OF DISPLAY REGISTER
                                                .WORD   DDISP
                                                .WORD   DDISP
                                                .WORD   DDISP
                                $$STP:          .WORD   STACK           ;;STACK INITIALIZATION FOR CPU0
                                                .WORD   STACK-2000      ;;''    ''                 '' CPU1
                                                .WORD   STACK-4000      ;;''    ''                 '' CPU2
                                                .WORD   STACK-10000     ;;''    ''                 '' CPU3
                                $TKS:           177560                  ;;TTY KBD STATUS
                                $TKB:           177562                  ;;TTY KBD BUFFER
                                $TPS:           177564                  ;;TTY PRINTER STATUS REG. ADDRESS
                                $TPB:           177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
                                $NULL:          .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
                                $FILLS:         .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
                                $FILLC:         .BYTE   12              ;;INSERT FILL CHARS. AFTER A 'LINE FEED''
                                $TPFLG:         .BYTE   0               ;;'TERMINAL AVAILABLE'' FLAG (BIT<07>=0-YES)
                                $ERGBL:         .WORD   0
                                $CPUID:         .WORD   -1              ;;THIS TABLE HOLDS THE PHYSICAL ID OF
                                                .WORD   -1              ;;THE PARTICIPATING PROCESSORS ARRANGED
                                                .WORD   -1              ;;IN LOGICAL ORDER.
                                                .WORD   -1
                                $REGAD:         .WORD   0       ;;CONTAINS THE ADDRESS FROM
                                                                ;;WHICH  ($REG0) WAS OBTAINED
                                $REG0:          .WORD   0,0,0,0         ;;CONTAINS (($REGAD)+0+6)

                                $REG1:          .WORD   0,0,0,0         ;;CONTAINS (($REGAD)+2+6)

                                $REG2:          .WORD   0,0,0,0         ;;CONTAINS (($REGAD)+4+6)

                                $REG3:          .WORD   0,0,0,0         ;;CONTAINS (($REGAD)+6+6)

                                $REG4:          .WORD   0,0,0,0         ;;CONTAINS (($REGAD)+10+6)
```

```
000000
000000                  $TMP0:  .WORD   0               ;;USER DEFINED
000000                  $TMP1:  .WORD   0               ;;USER DEFINED
000000                  $TMP2:  .WORD   0               ;;USER DEFINED
000000                  $TMP3:  .WORD   0               ;;USER DEFINED
000000                  $TMP4:  .WORD   0               ;;USER DEFINED
000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
000000                          .WORD   0
000000                          .WORD   0
000000                          .WORD   0
077                     $QUES:  .ASCII  /?/             ;;QUESTION MARK
015                     $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
                        ;;**************************************************************
                        .SBTTL  APT MAILBOX-ETABLE

                        ;;**************************************************************
                        .EVEN
014336                  $MAIL:                          ;;APT MAILBOX
000000                  $MSGTY: .WORD   AMSGTY  ;;MESSAGE TYPE CODE
000000                  $FATAL: .WORD   AFATAL  ;;FATAL ERROR NUMBER
000000                  $TESTN: .WORD   ATESTN  ;;TEST NUMBER
000000                  $PASS:  .WORD   APASS   ;;PASS COUNT
000000                  $DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
000000                  $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
000000                  $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
000000                  $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
014356                  $ETABLE:                        ;;APT ENVIRONMENT TABLE
000                     $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
000                     $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
000000                  $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
000000                  $USWR:  .WORD   AUSWR   ;;USER SWITCHES
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,PDQ=07,Q=10
                        ;*                      BIT 10=REAL TIME CLOCK
                        ;*                      BIT  9=FLOATING POINT PROCESSOR
                        ;*                      BIT  8=MEMORY MANAGEMENT
000                     $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
000                     $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
                        ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
                        ;*                              900 NSEC CORE=001
                        ;*                              300 NSEC BIPOLAR=002
                        ;*                              500 NSEC MOS=003
000000                  $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
                        ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF ''TYPE'' ABOVE
000                     $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
000                     $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
000000                  $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
000                     $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
000                     $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
000000                  $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
000                     $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
000                     $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
000000                  $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
```

```
                                        $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
                                        $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
                                        $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
                                        $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
                                        $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
                                        $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
                                        $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
                                        $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
                                        $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
                                        $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
                                        $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
                                        $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
                                        $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
                                        $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
                                        $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
                                        $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
                                        $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
                                        $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
                                        $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
                                        $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
                                        $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
                                        $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15

                                        $ETEND:

014462  000000  000000  000000  STOP:   0,0,0,0,0,0,0,0       ;MEM BOX UPPER BOUND ADDRESS TABLE
014470  000000  000000  000000
014476  000000  000000
014502  000000  000000  000000  START:  0,0,0,0,0,0,0,0       ;MEM BOX  STARTING ADDRESS TABLE
014510  000000  000000  000000
014516  000000  000000
014522  000000                  PWRFL:  0                       ; 0 DON'T EXPECT CPU POWER FAIL,=1 EXPECT IT
014524  000000                  YYY:    0                       ;ERROR ROUTINE WORK LOC
014526  000000                  BOOT:   0                       ;=0 DON'T EXPECT CPU BOOT, =1 EXPECT IT
014530  000000                  PATCHK: 0                       ;-0 WRITE AND CHECK MEM PATTERN,=1 CHECK ONLY
014532  000000                  HICORE: 0                       ;=1 TURN ON MM ON POWER-UP
014534  000000                  RELOUP: 0                       ;=0 DON'T RELOCATE,=1 RELOCATE
014536  000000                  RELODN: 0                       ;=0 DON'T RELOCATE,=1 RELOCATE
014540  000000                  EXIT:   0                       ;=0 DON'T EXIT, =1 EXIT TEST
014542  000000                  ENTR22: 0                       ;CONTROL ENTRY INTO TEST 22
014544  000000                  ENTR23: 0                       ;CONTROL ENTRY INTO TEST 23
014546  000000                  ENTR24: 0                       ;CONTROL ENTRY INTO TEST 24
014550  000000                  HIBOX:  0                       ;RELOCATE TO THIS MEM BOX
014552  000000  000000  000000  SAVRG:  0,0,0,0                 ;A PLACE TO SAVE A REGISTER
014560  000000
014562  000000  000000  000000  ROUTE:  0,0,0,0                 ;TYPE TRAP ROUTE TABLE
014570  000000

014572  000000  000000  000000  SAV6:   0,0,0,0                 ;SOME PLACE TO PUT THE SP
014600  000000
014602  000000  000000  000000  FLAG:   0,0,0,0                 ;INSTRUCTION DOWN FLAG
014610  000000
014612  000000  000000  000000  PFFT:   .WORD   0,0,0,0         ;;POWER FAIL FUNCTION TABLE
014620  000000
014622  000000  000000  000000  PFDT:   .WORD   0,0,0,0         ;;POWER FAIL DURATION TABLE
```

```
014630  000000
014632  000000  000000  000000  MBDSW:   .WORD   0,0,0,0          ;;MASSBUS DEVICE SELECTION WORD
014640  000000
014642  000000                  SIGNAL: 0                         ;POWER ROUTINE SIGNAL
014644  000000                  SLVID:   .WORD   0                ;;THIS WORD HOLDS ACTUAL LOG. ID.
014646  000000                  BOXNUM: .WORD    0                ;NUMBER OF BOXES OF MK11 MEMORY
014650  000000  000000  000000  CKSUM:   .WORD   0,0,0,0          ;CHECKSUM  TABLE
014656  000000
014660  000000                  $PSWR:   .WORD   0                ;PSEUDOSWITCH REGISTER
014662  000000  000000  000000  PWRTAB: .WORD    0,0,0,0          ;POWERFAIL DISPATCH TABLE
014670  000000
014672  000000  000000  000000  ISTTAB: .WORD    0,0,0,0          ;IIST DISPATCH TABLE
014700  000000
014702  000000  000000  000000  ERRTAB: .WORD    0,0,0,0          ;CPU ERROR VECTOR DISPATCH TABLE
014710  000000
014712  000000                  SYNC.1: .WORD    0                ;SEMAPHORE
014714  000000                  SYNC.2: .WORD    0                ;SEMAPHORE
014716  001000                  TYPQUE: .BLKW    1000             ;MESSAGE POINTER AREA
016716  000000                  S2LOG1: .WORD    0                ;LOG-IN LOCK
016720  000000                  S2LOG2: .WORD    0                ;LOG-OUT LOCK
016722  000000                  C1:      .WORD   0                ;;A-FORK CONTROL VARIABLES
016724  000000                  C2:      .WORD   0
016726  000000                  D1:      .WORD   0
016730  000000                  D2:      .WORD   0
016732  000000                  E1:      0
016734  000000                  E2:      0
016736     000                  FLAGB:   .BYTE   0
016737     000                  MPF:     .BYTE   0                ;MULTIPROCESSOR FLAG
016740  000000                  UBEF:    .WORD   0                ;UBE FLAB
016742  000000  000000  000000  RPPGM:   .WORD   0,0,0,0          ;SHARED RP04 DRIVE TABLE
016750  000000
016752  000000                  LOOPS:   .WORD   0                ;# OF LOOPS REQUIRED ON POWER DOWN
016754  000000                  COUNT0: 0                         ;TIME CPU POWER-DOWN
016756  000000                  COUNT1: 0
016760  000000                  COUNT2: 0
016762  000000                  COUNT3: 0
016764  000000                  EXTRA:   0                        ;EST. INITIAL PART OF PWR DWN TIME
016766  000001                  CPUACT: 1                         ;CPUS UNDER TEST
016770  000001                  SYNC.3: 1                         ;SEMAPHORE
016772  000001                  INTMSK: 1                         ;IIST INTERRUPT ENABLE BIT
016774  000001                  ERRLCK: 1                         ;ERROR ROUTINE SEMAPHORE
016776  000001                  C3:      1                        ;A-FORK CONTROL VARIABLES
017000  000001                  D3:      1
017002  000001                  E3:      1
017004  000001  000001  000001  NOPRMP: 1,1,1,1                   ;=1 DON'T TYPE CPU IDENTIFICATION
017012  000001
017014  000001                  UBELCK: 1                         ;UBE SEMAPHORE
017016  000001                  TQL1:    1                        ;TYPE SEMAPHORE
017020  000001                  S2L1:    .WORD   1                ;INITIALIZATION SEMAPHORE
017022  177500                  ACR:     177500                   ;IIST ACCESS CONTROL REGISTER
017024  177502                  ADR:     177502                   ;IIST ACCESS DATA REGISTER
017026  000260                  ISTVEC: 260                       ;IIST INTERRUPT VECTOR
017030  177777                  FIRST:   .WORD   -1               ;INCREMENTED BY EACH PROCESSOR
017032  000400                  BMSK:    .WORD   400              ;;IIST INITIAL BOOT MASK
017034  051052  061052  071052  BFADR:   .WORD   DSKBUF,DSKBUF+10000,DSKBUF+20000,DSKBUF+30000   ;MASSBUS TRANSFER BUF AR
017042  101052
```

```
  873  017044  000400  000400  000400  YELLIM: .WORD   400,400,400,400                    ;YELLOW ZONE BOUNDARY
  874  017052  000400
  875  017054  177777                  PUT:    -1                      ;CPU UNDER TEST
  876  017056  177777                  TYPLCK: -1                      ;ERROR/TYPE SEMAPHORE
  877  017060  006405                  FACTOR: 20000./6                ;POWER DWN FACTER
  878                                           .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 $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*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


017062                          $ERRTB:
                                        ;;ITEM  1
017062  045057                          EM1             ;UNEXPECTED POWER FAILURE ON CPU
017064  046346                          DH10            ;TESTNO  ERRORPC
017066  046752                          DT10            ;$TSTNM,$ERRPC
017070  000000                          0

                                        ;;ITEM  2
017072  045121                          EM2             ;UNEXPECTED POWER UP SEQUENCE ON CPU
017074  000000                          0
017076  000000                          0
017100  000000                          0

                                        ;;ITEM  3
017102  045167                          EM3             ;ILLEGAL POWER DOWN SEQUENCE
017104  000000                          0
017106  000000                          0
017110  000000                          0

                                        ;;ITEM  4
017112  045234                          EM4             ;ILLEGAL POWER UP SEQUENCE
017114  000000                          0
017116  000000                          0
017120  000000                          0

                                        ;;ITEM  5
017122  045270                          EM5             ;UNEXPECTED TRAP TO LOCATION 4
017124  046244                          DH5             ;PID    ERRORPC    CPUERR
017126  046730                          DT5             ;$REG0,$ERRPC,$REG1
017130  000000                          0

                                        ;;ITEM  6
017132  045317                          EM6             ;UNEXPECTED TRAP TO 10
017134  046244                          DH5             ;PID    ERRORPC    CPUERR
017136  046730                          DT5
017140  000000                          0

                                        ;;ITEM  7
017142  045347                          EM7             ;UNEXPECTED TRAP TO 114
017144  046277                          DH7             ;PID    ERRORPC    CPUERR    MEMERR
017146  046740                          DT7             ;$REG0,$ERRPC,$REG1,$REG2
017150  000000                          0
```

```
                                        ;;ITEM  10
017152  045400                          EM10            ;ADDRESS ON THE STACK IS WRONG
017154  046346                          DH10            ;TESTNO   ERRORPC
017156  046752                          DT10            ;$TSTNM,ERRORPC
017160  000000                          0

                                        ;;ITEM  11
017162  045434                          EM11            ;OLD PS IS WRONG
017164  046371                          DH11            ;TESTNO   ERRORPC    PS
017166  046760                          DT11            ;$TSTNM,$ERRPC,$REG0
017170  000000                          0

                                        ;;ITEM  12
017172  045456                          EM12            ;ODD ADDRESS TRAP FAILED
017174  046346                          DH10
017176  046752                          DT10
017200  000000                          0

                                        ;;ITEM  13
017202  045510                          EM13            ;MEMORY CORRUPTED ON POWER FAIL
017204  046423                          DH12
017206  046770                          DT12
017210  000000                          0

                                        ;;ITEM  14
017212  045551                          EM14            ;TIMEOUT TRAP FAILED
017214  046512                          DH14            ;TESTNO   ERRORPC    CPUERR
017216  046760                          DT11            ;$TSTNM,$ERRPC,$REG0
017220  000000                          0

                                        ;;ITEM  15
017222  045577                          EM15            ;POWER FAIL RETURNED TO SOON
017224  046346                          DH10
017226  046752                          DT10
017230  000000                          0

                                        ;;ITEM  16
017232  045636                          EM16            ;NOT ENOUGH OR TOO MABY INSTRUCTIONS EXECUTED
017234  046346                          DH10
017236  046752                          DT10
017240  000000                          0

                                        ;;ITEM  17
017242  045715                          EM17            ;NO MEM. MANG. VIOLATION OR TRAP TO 4
017244  046512                          DH14
017246  046760                          DT11
017250  000000                          0

                                        ;;ITEM  20
017252  045764                          EM20            ;NO IIST INTERRUPT
017254  046545                          DH20            ;TESTNO   ISTID   ACR   PGTE   PGCS
017256  047000                          DT20            ;$TSTNM,$REG0,$REG1,$REG2,$REG3
017260  000000                          0

                                        ;;ITEM  21
```

```
  991  017262  046010                    EM21            ;INCORRECT BRK AND/OR DCF FLAGS
  992  017264  046622                    DH21            ;TESTNO   ISTID   FOUND    SHOULD BE
  993  017266  047014                    DT21            ;$TSTNM,$REG0,$REG1,$REG2
  994  017270  000000                    0
  995
  996                                    ;;ITEM  22
  997  017272  046051                    EM22            ;CPU DID NOT TRAP TO 24
  998  017274  046672                    DH22            ;TESTNO  ISTID   ERRORPC
  999  017276  047026                    DT22            ;$TSTNM,$REG0,$ERRPC
 1000  017300  000000                    0
 1001
 1002                                    ;;ITEM  23
 1003  017302  046112                    EM23            ;CHECKSUM ON MASSBUS TRANSFER IS WRONG
 1004  017304  046672                    DH22
 1005  017306  047026                    DT22
 1006  017310  000000                    0
 1007
 1008                                    ;;ITEM  24
 1009  017312  046162                    EM24            ;NO POWER FAIL ON CPU
 1010  017314  046346                    DH10            ;TESTNO ERRORPC
 1011  017316  046752                    DT10            ;$TSTNM,$ERRPC
 1012  017320  000000                    0
 1013
 1014                                    ;;ITEM  25
 1015  017322  046211                    EM25            ;UNEXPECTED CPU INTERRUPT
 1016  017324  046346                    DH10            ;TESTNO ERRORPC
 1017  017326  046752                    DT10            ;$TSTNM,$ERRPC
 1018  017330  000000                    0
 1019
 1020
```

```
1021            020000                          .=20000                 ;;LOAD CODE ABOVE THE FIRST 4K (WORDS)
1022  020000                    RESTRT:
1023  020000  012704  016766            MOV     #CPUACT,        R4      ;RESET THE VARIABLES
1024  020004  012724  000001    1$:     MOV     #1,     (R4)+
1025  020010  022704  017020            CMP     #S2L1,  R4
1026  020014  103373                    BHIS    1$
1027  020016  012704  014462            MOV     #STOP,  R4
1028  020022  005024            2$:     CLR     (R4)+
1029  020024  022704  016764            CMP     #EXTRA, R4
1030  020030  103374                    BHIS    2$
1031  020032  012737  177777  017030    MOV     #-1,    FIRST
1032  020040  012737  000400  017032    MOV     #400,   BMSK
1033  020046  012737  177777  017074    MOV     #-1,    PUT
1034  020054  012737  177777  017056    MOV     #-1,    TYPLCK
1035  020062  000207                    RTS     PC                      ;RETURN
1036
1037  020064  013706  014200    STRT:   MOV     $$STP,  SP      ;INITIALIZE THE STACK
1038  020070  012704  014226            MOV     #$CPUID,R4              ;;GET ADDRESS OF CPUID TABLE
1039  020074  012703  000004            MOV     #4,R3                   ;;INIT COUNTER
1040  020100  012724  177777    4$:     MOV     #-1,(R4)+               ;;INIT TABLE WITH -1'S
1041  020104  077303                    SOB     R3,4$
1042                                    ;;INITIALIZE A FEW VECTORS
1043  020106  012737  036164  000030    MOV     #$ERROR,@#EMTVEC        ;;EMT VECTOR FOR ERROR ROUTINE
1044  020114  005037  000032            CLR     @#EMTVEC+2              ;;LEVEL 0
1045  020120  012737  040252  000034    MOV     #$TRAP,@#TRAPVEC        ;;TRAP VECTOR FOR TRAP CALLS
1046  020126  005037  000036            CLR     @#TRAPVEC+2             ;;LEVEL 0
1047  020132  012737  040340  000024    MOV     #PWRDIS,@#PWRVEC        ;;POINT TO POWER FAIL DISPATCH ROUTINE
1048  020140  005037  000026            CLR     @#PWRVEC+2              ;;LEVEL 0
1049  020144  012737  020330  000004    MOV     #25$,@#ERRVEC           ;;SET UP CPU ERROR VECTOR INCASE SWR IS'NT THERE
1050  020152  012737  014722  014716    MOV     #TYPQUE+4,TYPQUE        ;;INITIALIZE REAR POINTER
1051  020160  012737  014722  014720    MOV     #TYPQUE+4,TYPQUE+2      ;;INITIALIZE FORWARD POINTER
1052  020166  005000                    CLR     R0                      ;;SET ID IN DISPLACEMENT REGS.
1053  020170  005001                    CLR     R1
1054  020172  023737  000042  000046    CMP     @#42,   @#46            ;UNDER ACT AUTO MODE?
1055  020200  001402                    BEQ     40$                     ;BRANCH IF YES
1056  020202  104401  040376            TYPE    ,TM1                    ;TYPE PROGRAM NAME
1057  020206                    40$:
1058  020206  012737  177570  014160    MOV     #177570,        SWR     ;;SET SWR
1059  020214  132737  000200  014357    BITB    #APTSIZE,       $ENVM           ;;SIZE UNDER APT?
1060  020222  001403                    BEQ     50$                     ;;BRANCH IF NO
1061  020224  012737  014360  014160    MOV     #$SWREG,        SWR     ;;USE APT SWITCH REG
1062  020232  032777  000100  173720    BIT     #MPSW,@SWR      50$:    ;;IS HARDWARE SWITCH REG. THERE? AND MP SET?
1063  020240  001021                    BNE     20$                     ;;SWR IS THERE AND MP IS SET.
1064  020242  001435                    BEQ     30$                     ;;SWR IS THERE BUT MP IS NOT SET.
1065  020244  012737  040356  000004    MOV     #ERRDIS,@#ERRVEC        ;;RESET CPU ERROR VECTOR
1066  020252  012760  040362  014702    MOV     #CPUER,ERRTAB(R0)       ;;FLAG ALL UNEXPECTED TRAPS TO 4
1067  020260  012737  014360  014160    MOV     #$SWREG,SWR             ;;SETUP FOR SOFTWARE SWITCH REG
1068  020266  012737  000174  014170    MOV     #DISPREG,DISPLAY
1069  020274  032777  000100  173656    BIT     #MPSW,@SWR              ;;IS MP SWITCH SET IN SOFTWARE SWITCH REG?
1070  020302  001415                    BEQ     30$                     ;;NOPE.
1071  020304  012737  040356  000004    MOV     #ERRDIS,@#ERRVEC  20$:  ;;RESET ERROR VECTOR
1072  020312  012760  040362  014702    MOV     #CPUER,ERRTAB(R0)       ;;FLAG ALL UNEXPECTED TRAPS TO 4
1073  020320  152737  000001  016737    BISB    #BIT0,MPF               ;;SET THE MP FLAG.
1074  020326  000416                    BR      31$                     ;;LET'S GO
1075  020330  062716  000004    25$:    ADD     #4,(SP)                 ;;SKIP RETURN
1076  020334  000002                    RTI                             ;;RETURN FROM TRAP
```

```
 1077  020336  012737  040356  000004  30$:    MOV     #ERRDIS,@#ERRVEC        ;;RESET ERROR VECTOR
 1078  020344  012760  040362  014702          MOV     #CPUER,ERRTAB(R0)       ;;FLAG ALL UNEXPECTED TRAPS TO 4
 1079  020352  105037  016737                  CLRB    MPF                     ;;CLEAR THE MP FLAG
 1080  020356  104401  040663                  TYPE    ,TM7                    ;;[UNIPROCESSOR MODE IS IN EFFECT]
 1081  020362  000425                          BR      42$                     ;;ENTER INTO NON MP EXECUTION STREAM
 1082  020364  104401  040616          31$:    TYPE    ,TM6                    ;;[MULTIPROCESSOR MODE IS IN EFFECT]
 1083  020370  017737  173564  014660          MOV     @SWR,$PSWR              ;;SET UP PSEUDO SWITCH REGISTER.
 1084  020376  012704  014160                  MOV     #SWR,R4 ;;POINT TO SWR TABLE
 1085  020402  012705  000004                  MOV     #4,R5                   ;;SET COUNTER
 1086  020406  012724  014660          41$:    MOV     #$PSWR,(R4)+            ;;LOAD THE SLAVE SWITCH REG. POINTERS
 1087  020412  077503                          SOB     R5,41$                  ;;LOOP TILL DONE
 1088  020414  104401  040461                  TYPE    ,TM2                    ;;''SWITCH REGISTER = ''
 1089  020420  017746  173534                  MOV     @SWR,-(SP)              ;;SAVE @SWR FOR TYPEOUT
 1090  020424  104402                          TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 1091  020426  104401  014333                  TYPE    ,$CRLF                  ;;TYPE CRLF
 1092  020432  104401  014333                  TYPE    ,$CRLF
 1093  020436  032777  000040  173514  42$:    BIT     #UBESW,@SWR             ;;UBE SWITCH SET?
 1094  020444  001411                          BEQ     43$                     ;;NOT USED
 1095  020446  032777  000200  173504          BIT     #SW07,  @SWR            ;;WILL SECTION 1 BE SKIPPED?
 1096  020454  001005                          BNE     43$                     ;;BRANCH IF YES
 1097  020456  104401  040506                  TYPE    ,TM4                    ;;''[UNIBUS EXERCISER WILL BE USED]''
 1098  020462  105237  016740                  INCB    UBEF                    ;;SET UBE FLAG
 1099  020466  000404                          BR      65$
 1100  020470  104401  040550          43$:    TYPE    ,TM5                    ;;''[UNIBUS EXERCISER WILL NOT BE USED]''
 1101  020474  105037  016740                  CLRB    UBEF                    ;;CLEAR THE UBE FLAG
 1102  020500  105737  016737          65$:    TSTB    MPF                     ;;MULTIPROCESSOR MODE IN EFFECT?
 1103  020504  001002                          BNE     55$                     ;;BRANCH IF YES
 1104  020506  000137  021046                  JMP     ST0                     ;;START SETTING UP VECTORS
 1105                                  .SBTTL  BOOT AND INITIALIZE THE SLAVE CPUS
 1106  020512  017702  173442          55$:    MOV     @SWR,R2                 ;;COPY SWITCH REGISTER INTO R2
 1107  020516  012737  000001  016766          MOV     #1,CPUACT               ;;RESET # OF ACTIVE CPUS
 1108  020524  052777  100000  176270          BIS     #BIT15,@ACR             ;;INITIALIZE THE IIST.
 1109  020532  012777  000001  176262  81$:    MOV     #1,@ACR                 ;;ACCESS PGCS REGISTER
 1110  020540  032777  004000  176256          BIT     #BIT11,@ADR             ;;IS IT ''ALMOST READY''
 1111  020546  001771                          BEQ     81$                     ;;NOT YET.
 1112  020550  012705  002000                  MOV     #2000,R5                ;;SET UP COUNTER
 1113  020554  077501                          SOB     R5,.                    ;;WAIT UNTIL IIST IS REALLY READY
 1114  020556  017705  176240                  MOV     @ACR,R5                 ;;COPY ACR TO R5
 1115  020562  072527  177770                  ASH     #-10,R5                 ;;CREAT PHYSICAL ID
 1116  020566  104401  041450                  TYPE    ,TM76                   ;;IDENTIFY THE MASTER
 1117  020572  010546                          MOV     R5,     -(SP)
 1118  020574  104405                          TYPDS
 1119  020576  010537  014226                  MOV     R5,$CPUID               ;;SET SELF ID OF MASTER IN TABLE
 1120  020602  005000                          CLR     R0                      ;;REG0. CONTAINS WORD DISPLACE-
 1121                                                                          ;;MENT INTO CPUID TABLE ***
 1122  020604  005001                          CLR     R1                      ;;R1 CONTAINS THE BYTE DISPLACEMENT,...
 1123                                                                          ;;THE TRUE LOGICAL ID.
 1124  020606  012777  000007  176206          MOV     #DCF,@ACR               ;;ACCESS DCF REGISTER OF IIST
 1125  020614  017703  176204                  MOV     @ADR,R3 ;;COPY DCF INTO R3
 1126  020620  072327  177770                  ASH     #-10,R3                 ;;BRING BRK MASK INTO POSITION
 1127  020624  012777  021110  176174          MOV     #SLVENT,@ISTVEC         ;;SET ENTRY POINT FOR SLAVES
 1128
 1129  020632  005004                          CLR     R4                      ;;R4 CONTAINS THE DECIMAL VALUE
 1130                                                                          ;;OF THE SELF ID OF THE CPU UNDER
 1131                                                                          ;;INTEROGATION.
 1132  020634  032702  000001          1$:     BIT     #BIT0,R2                ;;DO WE WANT THIS CPU?
```

```
1133  020640  001462                          BEQ     2$              ;;NO,CONTINUE
1134  020642  032703  000001                  BIT     #BIT0,R3        ;;IS IT ALIVE?
1135  020646  001413                          BEQ     82$             ;;YES
1136  020650  010437  014310                  MOV     R4,$TMP0        ;;SAVE CONTENTS OF R4
1137  020654  104401  014333                  TYPE    ,$CRLF          ;;NO,CRLF
1138  020660  104401  040761                  TYPE    ,TM11           ;;''CPU # ''
1139  020664  013746  014310                  MOV     $TMP0,-(SP)     ;;SAVE $TMP0 FOR TYPEOUT
1140  020670  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
1141  020672  104401  040772                  TYPE    ,TM12           ;;''SPECIFIED BUT NOT ACTIVE''
1142  020676  020437  014226          82$:    CMP     R4,$CPUID       ;;IS THIS THE MASTER?
1143  020702  001441                          BEQ     2$              ;;YES, IGNORE
1144  020704  012737  020000  000000          MOV     #20000, @#0     ;M9312MP MOVES @#0 TO SP ON BOOT
1145  020712  012777  000000  176102          MOV     #PGTE,@ACR      ;;ACCESS PGTE REG.
1146  020720  013777  017032  176076          MOV     BMSK,@ADR       ;;SET TO BOOT AND THEN...
1147  020726  052777  000001  176070          BIS     #BIT0,@ADR      ;;BOOT THE CPU
1148  020734  012701  000200                  MOV     #200,   R1      ;;SET UP A LONG DELAY (5 SEC)
1149  020740  077001                  70$:    SOB     R0,     .
1150  020742  077102                          SOB     R1,     70$
1151  020744  012777  000001  176050  83$:    MOV     #PGCS,  @ACR    ;CHECK FOR IIST READY
1152  020752  032777  004000  176044          BIT     #BIT11, @ADR
1153  020760  001771                          BEQ     83$             ;BRANCH IF NOT
1154  020762  005077  176034          84$:    CLR     @ACR            ;;RESET ACR (POINT TO PGTE)
1155  020766  013777  016772  176030          MOV     INTMSK,@ADR     ;;SET UP TO INTERRUPT
1156  020774  052777  000001  176022          BIS     #BIT0,@ADR      ;;GO!,INTERRUPT SLAVE
1157  021002  005237  016766                  INC     CPUACT          ;COUNT ANOTHER ACTIVE CPU
1158  021006  020427  000003          2$:     CMP     R4,#3           ;;ALL CPUS STARTED?
1159  021012  002010                          BGE     3$              ;;YES
1160  021014  005204                          INC     R4              ;;NEXT CPU TO ATTEMPT TO BOOT
1161  021016  006202                          ASR     R2              ;;NEXT SWITCH REG. BIT
1162  021020  006203                          ASR     R3              ;;NEXT BRK BIT TO SET
1163  021022  006337  017032                  ASL     BMSK            ;;NEXT BOOT MASK
1164  021026  006337  016772                  ASL     INTMSK          ;;NEXT INTERRUPT MASK
1165  021032  000700                          BR      1$              ;;GO TRY ANOTHER
1166  021034  052737  001000  177746  3$:     BIS     #BIT9,  CONTRL  ;SET CACHE BYPASS
1167  021042  005037  000000                  CLR     @#0             ;RESTORE LOC. 0
1168
```

```
                                        .SBTTL  INITIALIZE THE COMMON TAGS


                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
021046  052737  000014  177746  STO:    BIS     #14,    CONTRL  ;DISABLE CACHE
021054  012706  014000                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
021060  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
021062  022706  014160                  CMP     #SWR,R6 ;;DONE?
021066  001374                          BNE     .-6             ;;LOOP BACK IF NO
021070  013706  014200                  MOV     $$STP,SP        ;;SETUP THE STACK POINTER
021074  013737  035200  035166          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
021102  005037  014344                  CLR     $PASS           ;;CLEAR PASS COUNT
021106  000432                          BR MSTENT
021110                          SLVENT:
021110  052737  001000  177746          BIS     #BIT9,@#CONTRL  ;;TURN OFF CACHE
021116  005237  014644                  INC     SLVID           ;;CREATE CPUID
021122  013701  014644                  MOV     SLVID,R1        ;;AND MOV TO R1
021126  010100                          MOV     R1,R0
021130  006300                          ASL     R0              ;;CREATE WORD INDEX INTO CPU TABLE
021132  017705  175664                  MOV     @ACR,   R5      ;;COPY ACR
021136  072527  177770                  ASH     #-10,   R5      ;;GET THE ID
021142  010560  014226                  MOV     R5,$CPUID(R0)   ;;SET SELF-ID INTO TABLE
021146  052777  100000  175646          BIS     #BIT15,@ACR     ;;RESET THE IIST
021154  012777  000001  175640          MOV     #PGCS,  @ACR    ;ENABLE INTERRUPTS
021162  052777  000004  175634          BIS     #BIT2,  @ADR
021170  005037  177776                  CLR     @#PSW           ;;LOWER PROCESSOR PRIORITY
021174                          MSTENT:
021174  005060  014322                  CLR     $ESCAPE(R0)             ;;CLEAR THE ESCAPE(R0) ON ERROR ADDRESS
021200  016060  021212  014032          MOV     10$(R0),$LPERR(R0)      ;;SETUP FOR THE ERROR LOOP ADDRESS
021206  000170  021222                  JMP     @FORKTB(R0)             ;;DISPATCH THE FOLLOWERS
021212  021420                  10$:    TST1
021214  021420                          TST1
021216  021420                          TST1
021220  021420                          TST1
021222  021312                  FORKTB: MS0
021224  021232                          SL1
021226  021252                          SL2
021230  021272                          SL3
021232  016006  014200          SL1:    MOV     $$STP(R0),SP    ;;INITIALIZE SLAVE STACK (CPU1)
021236  000001                          WAIT                    ;;WAIT FOR MASTER TO START VIA INTERRUPT
021240  052777  100000  175554          BIS     #BIT15,@ACR     ;;RESET THE IIST
021246  000137  021420                  JMP     TST1
021252  016006  014200          SL2:    MOV     $$STP(R0),SP    ;;INITIAL'ZE SLAVE STACK (CPU2)
021256  000001                          WAIT                    ;;WAIT FOR MASTER TO INTERRUPT
021260  052777  100000  175534          BIS     #BIT15,@ACR     ;;RESET THE IIST
021266  000137  021420                  JMP     TST1
021272  016006  014200          SL3:    MOV     $$STP(R0),SP    ;;INITIALIZE SLAVE STACK (CPU3)
021276  000001                          WAIT                    ;;WAIT FOR MASTER TO INTERRUPT
021300  052777  100000  175514          BIS     #BIT15,@ACR     ;;RESET THE IIST
021306  000137  021420                  JMP     TST1
021312  105737  016737          MS0:    TSTB    MPF             ;;MP MODE?
021316  001440                          BEQ     BEGIN           ;;NO,DON'T TRY TO INTERRUPT CPUS
021320  013702  017026                  MOV     ISTVEC, R2      ;SET UP RETURN FROM INTERRUPT
021324  062702  000002                  ADD     #2,     R2
021330  010277  175472                  MOV     R2,     @ISTVEC
```

```
 1225  021334  012712  000002                 MOV     #2,     (R2)
 1226  021340  012777  000000  175454         MOV     #PGTE,@ACR      ;ACCESS PGTE REGISTER
 1227  021346  017702  172606                 MOV     @SWR,R2         ;GET COPY OF SWR
 1228  021352  042702  177760                 BIC     #177760,R2      ;KEEP ONLY THE CPU MASK
 1229  021356  010277  175442                 MOV     R2,@ADR         ;SET THE INTERRUPT BITS
 1230  021362  032737  000001  016766         BIT     #BIT0,CPUACT    ;EVEN OR ODD?
 1231  021370  001404                         BEQ     7$              ;BRANCH IF EVEN
 1232  021372  012777  000005  175424         MOV     #5,@ADR         ;GO WITH NO PARITY,ENABLE INTERRUPTS
 1233  021400  000403                         BR      4$
 1234  021402  012777  000007  175414  7$:    MOV     #7,@ADR         ;GO WITH PARITY,ENABLE INTS.
 1235  021410  000001                 4$:     WAIT                    ;WAIT FOR IIST TO INTERRUPT
 1236  021412  052777  100000  175402         BIS     #BIT15,@ACR     ;;RESET THE IIST
 1237  021420                         BEGIN:
```

```
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255                    ;;**********************************************************************
1256                    ;;**********************************************************************
1257                    ;;**********                                          **********
1258                    ;;**********  S E C T I O N   O N E                   **********
1259                    ;;**********                                          **********
1260                    ;;**********************************************************************
1261                    ;;**********************************************************************
```

```
                                        ;;*************************************************************
                                        ;*TEST 1         SIMPLE DOWN/UP TEST (KERNAL)
                                        ;;*************************************************************
021420                                  TST1:
021420  012777  000001  172542                MOV     #1,@DISPLAY     ;SET TEST NUMBER
021426  012737  033530  014662                MOV     #POWDWN,        PWRTAB  ;SET UP POWER DOWN VECTOR
021434  012737  033074  000114                MOV     #PARERR,        @#CACHVEC       ;SET PARITY ERROR VECTOR
021442  012737  000001  033076                MOV     #1,     @#PARFLG        ;SET MULTI PARITY ERROR INDICATOR
021450  012737  040340  000024                MOV     #PWRDIS,@#PWRVEC        ;SET LOC 24
021456  005037  000026                        CLR     @#PWRVEC+2      ;SET LOC 26
021462  032770  000200  014160                BIT     #SW07,  @SWR(R0);SKIP SECTION 1?
021470  001402                                BEQ     1$              ;BRANCH IF NO
021472  000137  024170                        JMP     SEC2            ;ELSE GO TO SEC2
021476  005700                          1$:   TST     R0              ;IS THIS THE MASTER?
021500  001011                                BNE     4$              ;BRANCH IF NO
021502  104401  041370                        TYPE    ,TM14           ;'ENTERING SECTION 1''
021506  105737  016740                        TSTB    UBEF            ;USING THE UBE?
021512  001004                                BNE     4$              ;BRANCH IF YES
021514  104401  041024                        TYPE    ,TM13           ;PRINT INSTRUCTIONS
021520  104401  014333                        TYPE    ,$CRLF
021524                                  4$:
021524  005037  177776                        CLR     @#PS            ;SET KERNAL MODE
021530  012703  021566                        MOV     #2$,R3          ;SET POWER UP RETURN
021534  105737  016740                        TSTB    UBEF            ;;USE UNIBUS EXERCISER?
021540  001407                                BEQ     64$             ;;BRANCH IF NO
021542  106237  017014                        ASRB    UBELCK          ;LOCK OUT OTHER CPUS FROM PROCEEDING
021546  103375                              1 BCC     .-4
021550  000241                                CLC
021552  052737  000020  170016                BIS     #BIT4,@#UBCR2   ;;SET TO POWER FAIL
021560                                  64$:
021560                                  3$:
021560  005037  177776                        CLR     @#PS            ;SET KERNAL MODE
021564  000001                                WAIT                    ;WAIT FOR THE POWER FAIL
021566  010602                          2$:   MOV     SP,R2           ;GET SP
021570  016004  014200                        MOV     $$STP(R0),R4    ;R4 CONTAINS THE STACK INIT. VALUE
021574  162704  000004                        SUB     #4,R4           ;STACK-4
021600  020402                                CMP     R4,R2           ;CHECK STACK
021602  001401                                BEQ     .+4             ;SKIP IF OK
021604  000000                                HALT                    ;SP NOT 'STACK-4''
021606  016006  014200                        MOV     $$STP(R0),SP    ;RESET SP
021612  012402                                MOV     (R4)+,R2        ;GET RETURN ADDRESS
021614  105737  016740                        TSTB    UBEF            ;IS THE UBE BEING USED?
021620  001004                                BNE     72$             ;YES
021622  022702  021566                  70$:  CMP     #2$,R2          ;CHECK ADDRESS
021626  001401                                BEQ     .+4             ;SKIP IF OK
021630  000000                                HALT                    ;ADDRESS ON STACK IS WRONG
021632  011402                          72$:  MOV     (R4),R2         ;GET OLD PS
021634  022702  000000                        CMP     #0,R2           ;CHECK OLD PS
021640  001401                                BEQ     .+4             ;SKIP IF OK
021642  000000                                HALT                    ;OLD PS IS WRONG
021644  032770  040000  014160                BIT     #SW14,@SWR(R0)  ;LOOP ON TEST?
021652  001262                                BNE     TST1            ;LOOP TO TST1


                                        ;;*************************************************************
                                        ;*TEST 2         PROGRAM VOLATILITY TEST
```

```
                                        ;;*****************************************************************
021654                                  TST2:
021654  012777  000002  172306          MOV     #2,@DISPLAY     ;SET TEST NUMBER
021662  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
021666  012702  010000                  MOV     #10000,R2       ;INIT. COUNTER
021672  012703  020000                  MOV     #20000,R3       ;INIT POINTER
021676  005060  014650                  CLR     CKSUM(R0)       ;RESET THE CHECKSUM LOCATION
021702  062360  014650          1$:     ADD     (R3)+,CKSUM(R0) ;DO CHECKSUM ON 2ND 4K(W) BANK
021706  005560  014650                  ADC     CKSUM(R0)
021712  077205                          SOB     R2,1$
021714  012703  021750                  MOV     #2$,R3          ;POWER UP RETURN
021720  105737  016740                  TSTB    UBEF            ;;USE UNIBUS EXERCISER?
021724  001407                          BEQ     64$             ;;BRANCH IF NO
021726  106237  017014                  ASRB    UBELCK          ;LOCK OUT OTHER CPUS FROM PROCEEDING
021732  103375                          BCC     .-4
021734  000241                          CLC
021736  052737  000020  170016          BIS     #BIT4,@#UBCR2   ;;SET TO POWER FAIL
021744                          64$:
021744  000001                          WAIT                    ;WAIT FOR THE POWER TO FAIL
021746  000000                          HALT                    ;BAD
021750  012702  010000          2$:     MOV     #10000,R2
021754  012703  020000                  MOV     #20000,R3
021760  005004                          CLR     R4
                                        ;VERIFY THAT EVERYTHING IS OK
021762  062304                  3$:     ADD     (R3)+,R4
021764  005504                          ADC     R4
021766  077203                          SOB     R2,3$
021770  020460  014650                  CMP     R4,CKSUM(R0)    ;COMPARE NEW CHECKSUM WITH OLD
021774  001401                          BEQ     5$              ;BRANCH IF OK
021776  000000                          HALT                    ;ERROR
022000  016006  014200          5$:     MOV     $$STP(R0),SP    ;RESET THE STACK
022004  032770  040000  014160          BIT     #SW14,@SWR(R0)  ;LOOP ON TEST?
022012  001320                          BNE     TST2            ;LOOP TO TST2
                                        ;;*****************************************************************
                                        ;*TEST 3         SIMPLE DOWN/UP TEST (SUPERVISOR)
                                        ;;*****************************************************************
022014                                  TST3:
022014  012777  000003  172146          MOV     #3,@DISPLAY     ;SET TEST NUMBER
022022  012737  040000  177776          MOV     #40000,@#PS     ;SET SUPERVISOR MODE
022030  012703  022070                  MOV     #2$,R3          ;SET POWER UP RETURN
022034  105737  016740                  TSTB    UBEF            ;;USE UNIBUS EXERCISER?
022040  001407                          BEQ     64$             ;;BRANCH IF NO
022042  106237  017014                  ASRB    UBELCK          ;LOCK OUT OTHER CPUS FROM PROCEEDING
022046  103375                          BCC     .-4
022050  000241                          CLC
022052  052737  000020  170016          BIS     #BIT4,@#UBCR2   ;;SET TO POWER FAIL
022060                          64$:
022060                          3$:
022060  012737  040000  177776          MOV     #40000,@#PS     ;SET SUPERVISOR MODE
022066  000001                          WAIT                    ;WAIT FOR THE POWER FAIL
022070  016006  014200          2$:     MOV     $$STP(R0),SP    ;RESET
022074  016004  014200                  MOV     $$STP(R0),R4
022100  162704  000004                  SUB     #4,R4
```

```
1371  022104  012402                          MOV     (R4)+,R2        ;GET RETURN ADDRESS
1372  022106  105737  016740                  TSTB    UBEF
1373  022112  001004                          BNE     72$
1374  022114  022702  022070          70$:    CMP     #2$,R2          ;CHECK ADDRESS
1375  022120  001401                          BEQ     .+4             ;SKIP IF OK
1376  022122  000000                          HALT                    ;ADDRESS ON STACK IS WRONG
1377  022124  011402                  72$:    MOV     (R4),R2         ;GET OLD PS
1378  022126  022702  040000                  CMP     #40000,R2       ;CHECK OLD PS
1379  022132  001401                          BEQ     .+4             ;SKIP IF OK
1380  022134  000000                          HALT                    ;OLD PS IS WRONG
1381  022136                          1$:
1382  022136  032770  040000  014160          BIT     #SW14,@SWR(R0)  ;LOOP ON TEST?
1383  022144  001323                          BNE     TST3            ;LOOP TO TST3
1384
1385
1386                                  ;;**************************************************************
1387                                  ;*TEST 4         SIMPLE DOWN/UP TEST (USER)
1388                                  ;;**************************************************************
1389  022146                          TST4:
1390  022146  012777  000004  172014          MOV     #4,@DISPLAY     ;SET TEST NUMBER
1391  022154  012737  140000  177776          MOV     #140000,@#PS    ;SET USER MODE
1392  022162  012703  022222                  MOV     #2$,R3          ;SET POWER UP RETURN
1393  022166  105737  016740                  TSTB    UBEF            ;;USE UNIBUS EXERCISER?
1394  022172  001407                          BEQ     64$             ;;BRANCH IF NO
1395  022174  106237  017014                  ASRB    UBELCK          ;LOCK OUT OTHER CPUS FROM PROCEEDING
1396  022200  103375                          BCC     .-4
1397  022202  000241                          CLC
1398  022204  052737  000020  170016          BIS     #BIT4,@#UBCR2   ;;SET TO POWER FAIL
1399  022212                          64$:
1400  022212                          3$:
1401  022212  012737  140000  177776          MOV     #140000,@#PS    ;SET USER MODE
1402  022220  000001                          WAIT                    ;WAIT FOR THE POWER FAIL
1403  022222  016006  014200          2$:     MOV     $$STP(R0),SP    ;RESET SP
1404  022226  016004  014200                  MOV     $$STP(R0),R4    ;GET STACK INIT. VALUE
1405  022232  162704  000004                  SUB     #4,R4           ;MINUS 4
1406  022236  012402                          MOV     (R4)+,R2        ;GET STACK-4,AUTOINC. STACK
1407  022240  105737  016740                  TSTB    UBEF
1408  022244  001004                          BNE     72$
1409  022246  022702  022222          70$:    CMP     #2$,R2          ;CHECK ADDRESS
1410  022252  001401                          BEQ     .+4             ;SKIP IF OK
1411  022254  000000                          HALT                    ;ADDRESS ON STACK IS WRONG
1412  022256  011402                  72$:    MOV     (R4),R2         ;GET OLD PS
1413  022260  022702  140000                  CMP     #140000,R2      ;CHECK OLD PS
1414  022264  001401                          BEQ     .+4             ;SKIP IF OK
1415  022266  000000                          HALT                    ;OLD PS IS WRONG
1416  022270  032770  040000  014160          BIT     #SW14,@SWR(R0)  ;LOOP ON TEST?
1417  022276  001323                          BNE     TST4            ;LOOP TO TST4
1418
1419
1420
1421                                  ;;**************************************************************
1422                                  ;*TEST 5         POWER FAIL WITH ODD ADDRESS
1423                                  ;;**************************************************************
1424  022300                          TST5:
1425  022300  012777  000005  171662          MOV     #5,@DISPLAY     ;SET TEST NUMBER
1426  022306  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
```

```
 1427  022312  012760  022350  014702          MOV     #3$,ERRTAB(R0)  ;SET TRAP VECTOR
 1428  022320  012703  022376                  MOV     #1$,R3          ;SET RETURN ADDRESS FOR POWER FAIL
 1429  022324  105737  016740                  TSTB    UBEF            ;;USE UNIBUS EXERCISER?
 1430  022330  001407                          BEQ     64$             ;;BRANCH IF NO
 1431  022332  106237  017014                  ASRB    UBELCK          ;LOCK OUT OTHER CPUS FROM PROCEEDING
 1432  022336  103375                          BCC     .-4
 1433  022340  000241                          CLC
 1434  022342  052737  000020  170016          BIS     #BIT4,@#UBCR2   ;;SET TO POWER FAIL
 1435  022350                          64$:
 1436  022350  016006  014200          3$:     MOV     $$STP(R0),SP    ;RESET STACK
 1437  022354  005737  000003                  TST     @#3             ;CAUSE ODD ADDRESS TRAP
 1438  022360  105737  016740                  TSTB    UBEF            ;;USE UNIBUS EXERCISER?
 1439  022364  001403                          BEQ     65$
 1440  022366  042737  000020  170016          BIC     #BIT4,@#UBCR2   ;;CLEAR POWER FAIL ENABLE
 1441  022374                          65$:
 1442  022374  000000                          HALT                    ;ODD ADDRESS TRAP FAILED
 1443  022376  012760  040362  014702  1$:     MOV     #CPUER,ERRTAB(R0)       ;RESET 4
 1444  022404  032770  040000  014160          BIT     #SW14,@SWR(R0)  ;LOOP ON TEST?
 1445  022412  001332                          BNE     TST5            ;LOOP TO TST5
```

```
1446                                    ;;*****************************************************************
1447                                    ;*TEST 6        POWER FAIL IN THE RED ZONE
1448                                    ;;*****************************************************************
1449  022414                            TST6:
1450  022414  012777  000006  171546            MOV     #6,@DISPLAY     ;SET TEST NUMBER
1451  022422  005037  177776                    CLR     @#PS            ;SET KERNAL MODE
1452  022426  012760  022476  014702            MOV     #2$,ERRTAB(R0)  ;SET TRAP REGISTER
1453  022434  012703  022530                    MOV     #1$,R3          ;SET POWER UP RETURN
1454  022440  012706  000002                    MOV     #2,SP           ;SET STACK TO RED ZONE
1455  022444  105737  016740                    TSTB    UBEF            ;;USE UNIBUS EXERCISER?
1456  022450  001407                            BEQ     64$             ;;BRANCH IF NO
1457  022452  106237  017014                    ASRB    UBELCK          ;LOCK OUT OTHER CPUS FROM PROCEEDING
1458  022456  103375                            BCC     .-4
1459  022460  000241                            CLC
1460  022462  052737  000020  170016            BIS     #BIT4,@#UBCR2   ;;SET TO POWER FAIL
1461  022470                            64$:
1462  022470  005037  177776                    CLR     @#PS            ;SET KERNAL MODE
1463  022474  000001                            WAIT                    ;WAIT FOR POWER FAIL TRAP
1464  022476  012737  022504  014662  2$:       MOV     #7$,PWRTAB      ;SET UVEC TO HALT
1465  022504                            7$:
1466  022504  105737  016740                    TSTB    UBEF            ;;USE UNIBUS EXERCISER?
1467  022510  001403                            BEQ     65$
1468  022512  042737  000020  170016            BIC     #BIT4,@#UBCR2   ;;CLEAR POWER FAIL ENABLE
1469  022520                            65$:
1470  022520  000000                            HALT                    ;ILLEGAL TRAP TO 4
1471  022522  012737  033530  014662            MOV     #POWDWN,PWRTAB  ;RESET DVEC
1472  022530  016006  014200            1$:     MOV     $$STP(R0),SP    ;RESET STACK
1473  022534  012760  040362  014702            MOV     #CPUER,ERRTAB(R0)       ;RESET 4
1474  022542  013702  000002                    MOV     @#2,R2          ;GET FOR TYPING
1475  022546  023727  000002  000000            CMP     @#2,#0          ;IS 2 OK?
1476  022554  001401                            BEQ     .+4             ;SKIP IF OK
1477  022556  000000                            HALT                    ;NO!
1478  022560  013702  000000                    MOV     @#0,R2          ;GET FOR TYPING
1479  022564  022737  040340  000000            CMP     #PWRDIS,@#0     ;IS 0 OK?
1480  022572  001401                            BEQ     .+4             ;SKIP IF OK
1481  022574  000000                            HALT                    ;0 IS WRONG!
1482  022576  032770  040000  014160            BIT     #SW14,@SWR(R0)  ;LOOP ON TEST?
1483  022604  001303                            BNE     TST6            ;LOOP TO TST6
1484
1485
1486                                    ;;*****************************************************************
1487                                    ;*TEST 7        POWER FAIL WITH TIME OUT (KERNAL)
1488                                    ;;*****************************************************************
1489  022606                            TST7:
1490  022606  012777  000007  171354            MOV     #7,@DISPLAY     ;SET TEST NUMBER
1491  022614  005037  177776                    CLR     @#PS            ;SET KERNAL MODE
1492  022620  012760  022656  014702            MOV     #3$,ERRTAB(R0)  ;SET TRAP VECTOR
1493  022626  012703  022710                    MOV     #1$,R3          ;SET UP RETURN ADDRESS FOR POWER FAIL
1494  022632  105737  016740                    TSTB    UBEF            ;;USE UNIBUS EXERCISER?
1495  022636  001407                            BEQ     64$             ;;BRANCH IF NO
1496  022640  106237  017014                    ASRB    UBELCK          ;LOCK OUT OTHER CPUS FROM PROCEEDING
1497  022644  103375                            BCC     .-4
1498  022646  000241                            CLC
1499  022650  052737  000020  170016            BIS     #BIT4,@#UBCR2   ;;SET TO POWER FAIL
1500  022656                            64$:
1501  022656  016006  014200            3$:     MOV     $$STP(R0),SP    ;SET STACK
```

```
1502  022662  005037  177776                  CLR     @#PS              ;SET KERNAL MODE
1503  022666  010037  173000                  MOV     R0,@#173000       ;CAUSE A TIMEOUT
1504  022672  105737  016740                  TSTB    UBEF              ;;USE UNIBUS EXERCISER?
1505  022676  001403                          BEQ     65$
1506  022700  042737  000020  170016          BIC     #BIT4,@#UBCR2     ;;CLEAR POWER FAIL ENABLE
1507  022706                          65$:
1508  022706  000000                          HALT                      ;TIMEOUT FAILED
1509  022710  016006  014200          1$:     MOV     $$STP(R0),SP      ;SET STACK
1510  022714  012760  040362  014702          MOV     #CPUER,ERRTAB(R0)         ;RESET 4
1511  022722  032770  040000  014160          BIT     #SW14,@SWR(R0)    ;LOOP ON TEST?
1512  022730  001326                          BNE     TST7              ;LOOP TO TST7
1513
1514
1515                                  ;;*****************************************************************
1516                                  ;*TEST 10       POWER FAIL IN THE YELLOW ZONE (KERNAL)
1517                                  ;;*****************************************************************
1518  022732                          TST10:
1519  022732  012777  000010  171230          MOV     #10,@DISPLAY      ;SET TEST NUMBER
1520  022740  005037  177776                  CLR     @#PS              ;SET KERNAL MODE
1521  022744  005037  014602                  CLR     FLAG              ;CLEAR THE FLAG
1522  022750  012760  023040  014702          MOV     #2$,ERRTAB(R0)    ;SET SICK TPAP ADDRESS
1523  022756  012706  000400                  MOV     #400,SP           ;SET STACK TO YELLOW ZONE
1524  022762  012703  023020                  MOV     #1$,R3            ;SET RETURN ADDRESS FOR POWER FAIL
1525  022766  105737  016740                  TSTB    UBEF              ;;USE UNIBUS EXERCISER?
1526  022772  001407                          BEQ     64$               ;;BRANCH IF NO
1527  022774  106237  017014                  ASRB    UBELCK            ;LOCK OUT OTHER CPUS FROM PROCEEDING
1528  023000  103375                          BCC     .-4
1529  023002  000241                          CLC
1530  023004  052737  000020  170016          BIS     #BIT4,@#UBCR2     ;;SET TO POWER FAIL
1531  023012                          64$:
1532  023012  005037  177776                  CLR     @#PS              ;SET KERNAL MODE
1533  023016  000001                          WAIT                      ;WAIT FOR POWER FAIL
1534  023020                          1$:
1535  023020  105737  016740                  TSTB    UBEF              ;;USE UNIBUS EXERCISER?
1536  023024  001403                          BEQ     65$
1537  023026  042737  000020  170016          BIC     #BIT4,@#UBCR2     ;;CLEAR POWER FAIL ENABLE
1538  023034                          65$:
1539  023034  000000                          HALT                      ;POWER FAIL RETURNED TOO SOON
1540  023036  000430                          BR      4$                ;SKIP SP CHECK
1541  023040  012760  040362  014702  2$:     MOV     #CPUER,ERRTAB(R0)         ;RESET 4
1542  023046  005737  014602                  TST     FLAG              ;IS THE FIRST INSTRUCTION FLAG SET?
1543  023052  001016                          BNE     5$                ;YES
1544  023054  012737  023062  014662          MOV     #7$,PWRTAB        ;SET UVEC TO HALT
1545  023062                          7$:
1546  023062  105737  016740                  TSTB    UBEF              ;;USE UNIBUS EXERCISER?
1547  023066  001403                          BEQ     66$
1548  023070  042737  000020  170016          BIC     #BIT4,@#UBCR2     ;;CLEAR POWER FAIL ENABLE
1549  023076                          66$:
1550  023076  000000                          HALT                      ;NOT ENOUGH OR TOO MANY INSTR. EXEC.
1551  023100  012737  033530  014662          MOV     #POWDWN,PWRTAB    ;SET DVEC
1552  023106  000404                          BR      4$                ;GET OUT
1553  023110  012703  023120          5$:     MOV     #4$,R3            ;SET RETURN
1554  023114  000002                          RTI                       ;GO TO THE POWER FAIL ROUTINE
1555  023116  000000                          HALT                      ;SHOULD NOT RETURN HERE
1556  023120                          4$:
1557  023120  032770  040000  014160          BIT     #SW14,@SWR(R0)    ;LOOP ON TEST?
```

```
1558  023126  001301                          BNE     TST10           ;LOOP TO TST10
1559
1560
1561                                  ;;*****************************************************************
1562                                  ;*TEST 11       POWER FAIL WITH RESETS
1563                                  ;;*****************************************************************
1564  023130                          TST11:
1565  023130  012777  000011  171032          MOV     #11,@DISPLAY    ;SET TEST NUMBER
1566  023136  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
1567  023142  012703  023206                  MOV     #1$,R3          ;SET RETURN ADDRESS
1568  023146  016006  014200                  MOV     $$STP(R0),SP    ;RESET STACK
1569  023152  105737  016740                  TSTB    UBEF            ;;USE UNIBUS EXERCISER?
1570  023156  001407                          BEQ     64$             ;;BRANCH IF NO
1571  023160  106237  017014                  ASRB    UBELCK          ;LOCK OUT OTHER CPUS FROM PROCEEDING
1572  023164  103375                          BCC     .-4
1573  023166  000241                          CLC
1574  023170  052737  000020  170016          BIS     #BIT4,@#UBCR2   ;;SET TO POWER FAIL
1575  023176                          64$:
1576  023176  000005                  3$:     RESET                   ;RESETS
1577  023200  000005                          RESET                   ;TO WAIT
1578  023202  000005                          RESET                   ;IN
1579  023204  000774                          BR      3$              ;LOOP
1580  023206  016006  014200          1$:     MOV     $$STP(R0),SP    ;RESET STACK
1581  023212  032770  040000  014160          BIT     #SW14,@SWR(R0)  ;LOOP ON TEST?
1582  023220  001343                          BNE     TST11           ;LOOP TO TST11
1583
1584
1585                                  ;;*****************************************************************
1586                                  ;*TEST 12       POWER FAIL WITH ODD ADDRESS (SUPERVISOR)
1587                                  ;;*****************************************************************
1588  023222                          TST12:
1589  023222  012777  000012  170740          MOV     #12,@DISPLAY    ;SET TEST NUMBER
1590  023230  012737  040000  177776          MOV     #40000,@#PS     ;SET SUPERVISOR MODE
1591  023236  012760  023274  014702          MOV     #3$,ERRTAB(R0)  ;SET TRAP VECTOR
1592  023244  012703  023334                  MOV     #1$,R3          ;SET RETURN ADDRESS FOR POWER FAIL
1593  023250  105737  016740                  TSTB    UBEF            ;;USE UNIBUS EXERCISER?
1594  023254  001407                          BEQ     64$             ;;BRANCH IF NO
1595  023256  106237  017014                  ASRB    UBELCK          ;LOCK OUT OTHER CPUS FROM PROCEEDING
1596  023262  103375                          BCC     .-4
1597  023264  000241                          CLC
1598  023266  052737  000020  170016          BIS     #BIT4,@#UBCR2   ;;SET TO POWER FAIL
1599  023274                          64$:
1600  023274  016006  014200          3$:     MOV     $$STP(R0),SP    ;RESET STACK
1601  023300  012737  040000  177776          MOV     #40000,@#PS     ;SET SUPERVISOR MODE
1602  023306  005737  000003                  TST     @#3             ;CAUSE ODD ADDRESS TRAP
1603  023312  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
1604  023316  105737  016740                  TSTB    UBEF            ;;USE UNIBUS EXERCISER?
1605  023322  001403                          BEQ     65$
1606  023324  042737  000020  170016          BIC     #BIT4,@#UBCR2   ;;CLEAR POWER FAIL ENABLE
1607  023332                          65$:
1608  023332  000000                          HALT                    ;ODD ADDRESS TRAP FAILED
1609  023334  016006  014200          1$:     MOV     $$STP(R0),SP    ;RESET STACK POINTER
1610  023340  012760  040362  014702          MOV     #CPUER,ERRTAB(R0)       ;RESET 4
1611  023346  032770  040000  014160          BIT     #SW14,@SWR(R0)  ;LOOP ON TEST?
1612  023354  001322                          BNE     TST12           ;LOOP TO TST12
1613
```

```
1614
1615                                    ;;**********************************************************************
1616                                    ;*TEST 13       POWER FAIL WITH TIME OUT (SUPERVISOR)
1617                                    ;;**********************************************************************
1618  023356                            TST13:
1619  023356  012777  000013  170604            MOV     #13,@DISPLAY    ;SET TEST NUMBER
1620  023364  012737  040000  177776            MOV     #40000,@#PS     ;SET SUPERVISOR MODE
1621  023372  012760  023430  014702            MOV     #3$,ERRTAB(R0)  ;SET TRAP VECTOR
1622  023400  012703  023470                    MOV     #1$,R3          ;SET UP RETURN ADDRESS FOR POWER FAIL
1623  023404  105737  016740                    TSTB    UBEF            ;;USE UNIBUS EXERCISER?
1624  023410  001407                            BEQ     64$             ;;BRANCH IF NO
1625  023412  106237  017014                    ASRB    UBELCK          ;LOCK OUT OTHER CPUS FROM PROCEEDING
1626  023416  103375                            BCC     .-4
1627  023420  000241                            CLC
1628  023422  052737  000020  170016            BIS     #BIT4,@#UBCR2   ;;SET TO POWER FAIL
1629  023430                            64$:
1630  023430  016006  014200            3$:     MOV     $$STP(R0),SP    ;RESET STACK
1631  023434  012737  040000  177776            MOV     #40000,@#PS     ;SET SUPERVISOR MODE
1632  023442  010037  173000                    MOV     R0,@#173000     ;CAUSE A TIMEOUT
1633  023446  005037  177776                    CLR     @#PS            ;SET KERNAL MODE
1634  023452  105737  016740                    TSTB    UBEF            ;;USE UNIBUS EXERCISER?
1635  023456  001403                            BEQ     65$
1636  023460  042737  000020  170016            BIC     #BIT4,@#UBCR2   ;;CLEAR POWER FAIL ENABLE
1637  023466                            65$:
1638  023466  000000                            HALT                    ;TIMEOUT FAILED
1639  023470  016006  014200            1$:     MOV     $$STP(R0),SP    ;RESET STACK
1640  023474  012760  040362  014702            MOV     #CPUER,ERRTAB(R0)       ;RESET 4
1641  023502  032770  040000  014160            BIT     #SW14,@SWR(R0)  ;LOOP ON TEST?
1642  023510  001322                            BNE     TST13           ;LOOP TO TST13
1643
1644
1645                                    ;;**********************************************************************
1646                                    ;*TEST 14       POWER FAIL WITH ODD ADDRESS (USER)
1647                                    ;;**********************************************************************
1648  023512                            TST14:
1649  023512  012777  000014  170450            MOV     #14,@DISPLAY    ;SET TEST NUMBER
1650  023520  012737  140000  177776            MOV     #140000,@#PS    ;SET USER MODE
1651  023526  012760  023564  014702            MOV     #3$,ERRTAB(R0)  ;SET TRAP VECTOR
1652  023534  012703  023624                    MOV     #1$,R3          ;SET RETURN ADDRESS FOR POWER FAIL
1653  023540  105737  016740                    TSTB    UBEF            ;;USE UNIBUS EXERCISER?
1654  023544  001407                            BEQ     64$             ;;BRANCH IF NO
1655  023546  106237  017014                    ASRB    UBELCK          ;LOCK OUT OTHER CPUS FROM PROCEEDING
1656  023552  103375                            BCC     .-4
1657  023554  000241                            CLC
1658  023556  052737  000020  170016            BIS     #BIT4,@#UBCR2   ;;SET TO POWER FAIL
1659  023564                            64$:
1660  023564  016006  014200            3$:     MOV     $$STP(R0),SP    ;RESET STACK
1661  023570  012737  140000  177776            MOV     #140000,@#PS    ;SET USER MODE
1662  023576  005737  000003                    TST     @#3             ;CAUSE ODD ADDRESS TRAP
1663  023602  005037  177776                    CLR     @#PS            ;SET KERNAL MODE
1664  023606  105737  016740                    TSTB    UBEF            ;;USE UNIBUS EXERCISER?
1665  023612  001403                            BEQ     65$
1666  023614  042737  000020  170016            BIC     #BIT4,@#UBCR2   ;;CLEAR POWER FAIL ENABLE
1667  023622                            65$:
1668  023622  000000                            HALT                    ;ODD ADDRESS TRAP FAILED
1669  023624  016006  014200            1$:     MOV     $$STP(R0),SP    ;RESET SP
```

```
                                MOV     #CPUER,ERRTAB(R0)       ;RESET 4
                                BIT     #SW14,@SWR(R0)  ;LOOP ON TEST?
                                BNE     TST14           ;LOOP TO TST14


                        ;;**********************************************************************
                        ;*TEST 15       POWER FAIL WITH TIME OUT (USER)
                        ;;**********************************************************************
                        TST15:
                                MOV     #15,@DISPLAY    ;SET TEST NUMBER
                                MOV     #140000,@#PS    ;SET USER MODE
                                MOV     #3$,ERRTAB(R0)  ;SET TRAP VECTOR
                                MOV     #1$,R3          ;SET UP RETURN ADDRESS FOR POWER FAIL
                                TSTB    UBEF            ;;USE UNIBUS EXERCISER?
                                BEQ     64$             ;;BRANCH IF NO
                                ASRB    UBELCK          ;LOCK OUT OTHER CPUS FROM PROCEEDING
                                BCC     .-4
                                CLC
                                BIS     #BIT4,@#UBCR2   ;;SET TO POWER FAIL
                        64$:
                        3$:     MOV     $$STP(R0),SP    ;RESET STACK
                                MOV     #140000,@#PS    ;SET USER MODE
                                MOV     R0,@#173000     ;CAUSE A TIMEOUT
                                CLR     @#PS            ;SET KERNAL MODE
                                TSTB    UBEF            ;;USE UNIBUS EXERCISER?
                                BEQ     65$
                                BIC     #BIT4,@#UBCR2   ;;CLEAR POWER FAIL ENABLE
                        65$:
                                HALT                    ;TIMEOUT FAILED
                        1$:     MOV     $$STP(R0),SP    ;RESET STACK
                                MOV     #CPUER,ERRTAB(R0)       ;RESET 4
                                BIT     #SW14,@SWR(R0)  ;LOOP ON TEST?
                                BNE     TST15           ;LOOP TO TST15


                        ;;**********************************************************************
                        ;*TEST 16       MEMORY MANAGEMENT ABORT TEST
                        ;;**********************************************************************
                        TST16:
                                MOV     #16,@DISPLAY    ;SET TEST NUMBER
                                CLR     @#PS            ;SET KERNAL MODE
                                MOV     #TI,    PFFT(R0)        ;TIME THIS POWER FAIL
                                MOV     #4$,ERRTAB(R0)  ;SET FOR TIMEOUT
                                JSR     PC,MAP          ;MAP THE WORLD
                                MOV     #3$,@#MMVEC     ;SET MEMORY MANAGEMENT VECTOR
                                MOV     #1$,R3          ;LOAD PF RETURN
                                INC     @#MMR0          ;TURN MEMORY MANAGEMENT ON
                                TSTB    UBEF            ;;USE UNIBUS EXERCISER?
                                BEQ     64$             ;;BRANCH IF NO
                                ASRB    UBELCK          ;LOCK OUT OTHER CPUS FROM PROCEEDING
                                BCC     .-4
                                CLC
                                BIS     #BIT4,@#UBCR2   ;;SET TO POWER FAIL
                        64$:
                        3$:     MOV     $$STP(R0),SP    ;ZAP STACK
                                INC     @#140000        ;ACCESS VIOLATION
```

```
 1726  024106  105737  016740                  TSTB    UBEF            ;;USE UNIBUS EXERCISER?
 1727  024112  001403                          BEQ     65$
 1728  024114  042737  000020  170016          BIC     #BIT4,@#UBCR2   ;;CLEAR POWER FAIL ENABLE
 1729  024122                          65$:
 1730  024122  000000                  4$:     HALT                    ;NO VIOLATION OR TRAP TO 4
 1731
 1732  024124  005037  177572          1$:     CLR     @#MMR0          ;TURN OFF MEMORY MANAGEMENT
 1733  024130  016006  014200          2$:     MOV     $$STP(R0),SP    ;MAKE A NEW STACK
 1734  024134  012760  040362  014702          MOV     #CPUER,ERRTAB(R0)       ;RESET 4
 1735  024142  032770  040000  014160          BIT     #SW14,@SWR(R0)  ;LOOP ON TEST?
 1736  024150  001314                          BNE     TST16           ;LOOP TO TST16
 1737  024152  005077  170012                  CLR     @DISPLA         ;CLEAR THE DISPLAY REGISTER.
 1738  024156  105737  016737                  TSTB    MPF             ;MP MODE?
 1739  024162  001002                          BNE     5$              ;BRANCH IF YES
 1740  024164  000137  035060                  JMP     $EOP            ;UUMP INTO EOP
 1741  024170                          5$:
```

```
1742
```

```
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759                    ;*************************************************************************
1760                    ;*************************************************************************
1761                    ;******************                                     **************
1762                    ;****************** S E C T I O N  T W O               **************
1763                    ;******************                                     **************
1764                    ;*************************************************************************
1765                    ;*************************************************************************
```

```
                                        .SBTTL  SECTION 2 INITIALIZATION
024170  012703  024214          SEC2:   MOV     #1$,    R3              ;SET UP POWER FAIL RETURN IN CASE
024174  106237  016770                  ASRB    SYNC.3                  ;CONTROL THE ENTRY
024200  103373                          BCC     SEC2
024202  005237  016716                  INC     S2LOG1                  ;LOG INTO SEC2
024206  012737  000001  016770          MOV     #1,     SYNC.3          ;LET THE OTHERS IN
024214  023737  016716  016766  1$:     CMP     S2LOG1,CPUACT           ;;WAIT FOR EVERYONE TO GET ...
024222  001374                          BNE     1$                      ;;HERE
024224  005700                          TST     R0                      ;;IS THIS THE MASTER?
024226  001002                          BNE     3$                      ;;BRANCH IF NO
024230  104401  041420                  TYPE    ,TM15                   ;;'ENTERING SECTION 2''
024234                          3$:
024234  052777  100000  172560          BIS     #BIT15, @ACR            ;INITIALIZE IIST
024242  016006  014200                  MOV     $$STP(R0),      SP      ;SET THE STACK
024246  106237  017020          2$:     ASRB    S2L1                    ;;TRY TO ENTER SECTION 2 INITIALIZATION
024252  103375                          BCC     2$
024254  012737  034220  000024          MOV     #$POWER,@#24            ;;SET NEW POWER FAIL HANDLER
024262  012737  000340  000026          MOV     #340,@#26
024270  017705  172526                  MOV     @ACR,   R5      ;COPY ACR
024274  072527  177770                  ASH     #-10,   R5      ;GET THE ID
024300  010560  014240                  MOV     R5,     $REG0(R0);IDENTIFY CPU FOR ERROR TYPE-OUT
024304  005002                          CLR     R2              ;RESET FOR COUNT
024306  026205  014226          65$:    CMP     $CPUID(R2),R5   ;SID MATCH?
024312  001404                          BEQ     64$
024314  005722                          TST     (R2)+           ;INCREMENT R2 BY 2
024316  020227  000010                  CMP     R2,#10
024322  002771                          BLT     65$
024324  010200          64$:    MOV     R2,R0           ;MOV LOGICAL ID TO 2ND OPERAND
024326  010001                  MOV     R0,     R1              ;SET UP R1
024330  006201                  ASR     R1
024332  012760  034412  014662          MOV     #$PWRDN,        PWRTAB(R0)      ;SET UP FOR POWER DOWN
024340  012760  024634  014702  SIZMBS: MOV     #NORP,ERRTAB(R0)                ;;SET UP CPU ERROR VECTOR IN CASE
                                                                        ;;THERE'S NO RP0/5/6
024346  052737  000040  176710          BIS     #BIT5,@#RPCS2           ;;INIT. RP CONTROLLER,IF THERE.
024354  012760  040362  014702          MOV     #CPUER,ERRTAB(R0)       ;;RESET ERROR VECTOR
024362  005002                          CLR     R2                      ;;RESET COUNTER
024364  010237  176710          RPSRC:  MOV     R2,@#RPCS2              ;;SET DRIVE # IN CS REG.
024370  032737  040000  176712          BIT     #BIT14,@#RPDS           ;;IS THE DRIVE UP?
024376  001024                          BNE     NXTDRV                  ;;BRANCH IF IT ISN'T

024400  032737  001000  176712          BIT     #BIT9,@#RPDS            ;;IS THE PGM BIT SET FOR THIS DRIVE
024406  001403                          BEQ     1$
024410  005260  016742                  INC     RPPGM(R0)               ;;YES, FLAG THE CONDITION
024414  000415                          BR      NXTDRV                  ;;AND, SEARCH FOR ANOTHER DRIVE
024416  032737  000400  176712  1$:     BIT     #BIT8,@#RPDS            ;;IS THIS PORT IN CONTROL?
024424  001411                          BEQ     NXTDRV                  ;;NO, LOOK FOR ANOTHER
024426  005060  014632                  CLR     MBDSW(R0)               ;;CLEAR MASSBUSS DEVICE SELECTION
024432  013703  176710                  MOV     @#RPCS2,R3              ;;COPY CS2
024436  042703  177770                  BIC     #177770,R3              ;;GET RID OF OTHER INFO
024442  110360  014632                  MOVB    R3,MBDSW(R0)            ;;WRITE DRIVE ID INTO SELECTION W
024446  000454                          BR      SIZEND                  ;;DEVICE HAS BEEN FOUND.
024450  005202          NXTDRV: INC     R2                      ;;NEXT DRIVE
024452  020227  000010                  CMP     R2,#10                  ;;ALL DRIVES TESTED?
024456  103742                          BLO     RPSRC                   ;;NO, TEST SOME MORE.
024460  005760  016742                  TST     RPPGM(R0)               ;;ANY PROGRAMMABLE DRIVES?
024464  001011                          BNE     A                       ;;YES.
```

```
1822  024466  104401  041533          NORH70: TYPE    ,TM101                  ;;'NO MASSBUS DEVICE AVAILABLE ON CPU #''
1823  024472  016046  014226                  MOV     $CPUID(R0),-(SP)        ;;SAVE $CPUID(R0) FOR TYPEOUT
1824  024476  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
1825  024500  012760  001400  014632          MOV     #1400,MBDSW(R0)         ;;SET CODE FOR NO DEVICE
1826  024506  000434                          BR      SIZEND                  ;;EXIT SECTION 2 INITIALIZATION
1827  024510  005002                  A:      CLR     R2                      ;;RESET COUNTER
1828  024512  010237  176710          10$:    MOV     R2,@#RPCS2              ;;ACCESS DRIVE.
1829  024516  032737  001000  176712          BIT     #BIT9,@#RPDS            ;;PGM BIT SET?
1830  024524  001005                          BNE     15$                     ;;YES, FOUND ONE
1831  024526  005202                          INC     R2                      ;;NO, NEXT DRIVE
1832  024530  020227  000010                  CMP     R2,#10                  ;;ALL DRIVE TESTED
1833  024534  103766                          BLO     10$                     ;;NO.
1834  024536  000000                          HALT                            ;;YES.
1835  024540  110260  014632          15$:    MOVB    R2,MBDSW(R0)            ;;SET DRIVE #
1836  024544  052760  001000  014632          BIS     #BIT9,MBDSW(R0)         ;;SET PGMBIT
1837  024552  104401  043535                  TYPE    ,$PGM1                  ;TYPE SHARED DRIVE WARNING MSG
1838  024556  010246                          MOV     R2,     -(SP)
1839  024560  104405                          TYPDS
1840  024562  104401  043563                  TYPE    ,$PGM2
1841  024566  016046  014226                  MOV     $CPUID(R0),     -(SP)
1842  024572  104405                          TYPDS
1843  024574  104401  043577                  TYPE    ,$PGM3
1844  024600  012737  000001  017020  SIZEND: MOV     #1,S2L1                 ;;ALLOW ENTRY INTO SEC. 2
1845  024606  005237  016720                  INC     S2LOG2                  ;;LOG OUT OF SECTION 2 INITIALIZATION
1846  024612  023737  016766  016720  1$:     CMP     CPUACT,S2LOG2           ;WAIT FOR ALL THE CPUS TO GET HERE
1847  024620  001374                          BNE     1$
1848  024622  012777  040334  172176          MOV     #ISTDIS,@ISTVEC         ;;POINT TO DISPATCHER
1849  024630  000137  024644                  JMP     TST17                   ;START THE TEST
1850
1851  024634  000240                  NORP:   NOP                             ;THERE IS NO RP CONTROLLER
1852  024636  012716  024466                  MOV     #NORH70,(SP)            ;;SET FOR TEST ENTRY RETURN
1853  024642  000002                          RTI                             ;;RETURN
1854
1855
1856
1857
1858
```

```
                                        ;;*************************************************************
                                        ;*TEST 17       CHECK 'BRK' & 'DCF' FLAGS DURING POWERFAIL
                                        ;;*************************************************************
024644                                  TST17:
024644  012777  000017  167316          MOV     #17,@DISPLAY    ;SET TEST NUMBER

                                        ;;***************** TS17A-FORK ***************************

024652  016006  014200                  MOV     $$STP(R0),      SP;SET UP THE STACK
024656  052777  100000  172136          BIS     #BIT15, @ACR    ;INITIALIZE THE IIST
024664  106277  000122          70$:    ASRB    @69$            ;ENTER CONTROL FORK
024670  103375                          BCC     70$
024672  027737  000110  016766          CMP     @67$,CPUACT             ;HAVE WE REACHED THE END OF THE
                                        ;ROUTING CYCLE?
024700  001021                          BNE     65$             ;BRANCH IF NO
024702  013702  016766                  MOV     CPUACT,R2
024706  005302                          DEC     R2
024710  006302                          ASL     R2              ;(CPUACT-1)*2
024712  027702  000072                  CMP     @68$,R2         ;ARE WE AT THE END OF THE TEST?
024716  001005                          BNE     64$             ;BRANCH IF NO
024720  012777  000001  000064          MOV     #1,     @69$    ;EXIT
024726  000137  025362                  JMP     TST20
024732  062777  000002  000050  64$:    ADD     #2,     @68$            ;INCREMENT 68$ BY 2
024740  005077  000042                  CLR     @67$            ;CLEAR THE CHECKPOINT COUNTER
024744  005277  000036          65$:    INC     @67$            ;INCREMENT CHECKPOINT
024750  005037  014714                  CLR     SYNC.2          ;CLEAR THE LOCK
024754  020077  000030                  CMP     R0,@68$         ;ROUTE THIS PROCESSOR THROUGH TS17A?
024760  001005                          BNE     66$             ;BRANCH IF NO
024762  012777  000001  000022          MOV     #1,@69$         ;CLEAR LOCK
024770  000137  025014                  JMP     TS17A           ;JUMP TO BRANCH TS17A
024774  012777  000001  000010  66$:    MOV     #1,@69$         ;CLEAR LOCK
025002  000137  025136                  JMP     TS17B           ;JUMP TO TS17B
                                        ;;*************************************************************
025006  016722                  67$:    C1
025010  016724                  68$:    C2
025012  016776                  69$:    C3

025014  112761  000017  014002  TS17A:  MOVB    #17,    $TSTNM(R1)      ;SET UP THE TEST NUMBER
025022  010037  017054                  MOV     R0,PUT          ;SET PROCESSOR UNDER TEST
025026  005700                          TST     R0              ;IS THIS THE MASTER?
025030  001007                          BNE     5$              ;BRANCH IF NO
025032  104401  041476                  TYPE    ,TM77           ;'TEST'
025036  005046                          CLR     -(SP)
025040  116116  014002                  MOVB    $TSTNM(R1),(SP) ;GET THE TEST NO.
025044  104403                          TYPOS
025046  000002                          .WORD   2                       ;TYPE 2 DIGITS, NO LEADING 0
025050                          5$:
025050  104401  041507                  TYPE    ,TM100          ;'POWER FAIL CPU #'
025054  016046  014226                  MOV     $CPUID(R0),-(SP)        ;;SAVE $CPUID(R0) FOR TYPEOUT
025060  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
025062  104401  014333                  TYPE    ,$CRLF
025066  012760  014020  014612          MOV     #SSU!TI!NCX,PFFT(R0)    ;SEND SIG. ON UP, TIME, DON'T SAVE MM
025074  012760  025124  014672          MOV     #BAD,ISTTAB(R0) ;SET UP IIST VECTOR FOR THIS CPU.
025102  005003                          CLR     R3              ;RETURN AFTER THE WAIT ON POWER UP
025104  005037  014642                  CLR     SIGNAL                  ;CLEAR THE POWER UP SIGNAL
025110  012737  177777  014712          MOV     #-1,    SYNC.1          ;UNLOCK THE OTHER CPUS
```

```
1915  025116  000001                          WAIT                    ;WAIT FOR THE POWER TO FAIL.
1916  025120  000137  024644                  JMP     TST17           ;GO TO CONTROL FORK
1917
1918
1919  025124  104025                  BAD:    ERROR   25              ;UNEXPECTED CPU INTERRUPT
1920  025126  000000                          HALT
1921  025130  012716  024644                  MOV     #TST17, (SP)    ;CONTINUE TESTING
1922  025134  000002                          RTI
1923
1924  025136  112761  000017  014002  TS17B:  MOVB    #17,    $TSTNM(R1)      ;SET UP THE TEST NUMBER
1925  025144  106237  014712                  ASRB    SYNC.1          ;WAIT FOR P.U.T. TO SEND SIGNAL
1926  025150  103372                          BCC     TS17B           ;LOOP UNTIL SENT
1927  025152  104401  014220                  TYPE    ,$NULL          ;FLUSH THE TYPE QUEUE
1928  025156  012760  025124  014672          MOV     #BAD,   ISTTAB(R0)      ;GET SET FOR BAD INTERRUPT
1929  025164  012777  000001  171630          MOV     #PGCS,@ACR      ;ACCESS PGCS REGISTER
1930  025172  052777  000004  171624          BIS     #BIT2,@ADR      ;SET THE INTERRUPT ENABLE BIT
1931
1932  025200  012777  000007  171614          MOV     #DCF,@ACR       ;COPY THE DCF REG.
1933  025206  017705  171612                  MOV     @ADR,R5 ;INTO R5
1934  025212  012760  025264  014672          MOV     #STS17, ISTTAB(R0)      ;SET FOR EXPECTED INTERRUPT
1935  025220  106237  014642          1$:     ASRB    SIGNAL          ;WAIT ON POWER FAIL SIGNAL
1936  025224  103375                          BCC     1$              ;
1937  025226  017760  171570  014250          MOV     @ACR,$REG1(R0)  ;SAVE THE ACR
1938  025234  012777  000000  171560          MOV     #PGTE,@ACR      ;ACCESS THE PGTE REG.
1939  025242  017760  171556  014260          MOV     @ADR,$REG2(R0)  ;SAVE THE PGTE REG.
1940  025250  017760  171550  014270          MOV     @ADR,$REG3(R0)  ;SAVE THE PGCS REG.
1941  025256  104020                          ERROR   20              ;NO IIST INTERRUPT
1942  025260  000137  024644                  JMP     TST17           ;DO IT AGAIN
1943  025264                          STS17:
1944  025264  012777  000007  171530          MOV     #DCF,@ACR       ;ACCESS DCF REGISTER
1945  025272  013703  017054                  MOV     PUT,R3          ;GET LOGICAL ID INTO R3
1946  025276  016304  014226                  MOV     $CPUID(R3),R4   ;COPY IIST ID TO DESTINATION.
1947  025302  012703  000401                  MOV     #401,R3         ;MAKE A MASK
1948  025306  072304                          ASH     R4,R3           ;BRING IT INTO POSITION
1949  025310  050305                          BIS     R3,R5           ;R5 IS WHAT THE DCF REG. SHOULD LOOK LIKE
1950  025312  077301                          SOB     R3,     .       ;DELAY A SHORT WHILE
1951  025314  017702  171504                  MOV     @ADR,R2 ;COPY DCF REGISTER
1952  025320  020205                          CMP     R2,R5           ;EVERYTHING OK?
1953  025322  001405                          BEQ     3$              ;BRANCH IF YES
1954  025324  010260  014250                  MOV     R2,$REG1(R0)    ;THE DCF REG.
1955  025330  010560  014260                  MOV     R5,$REG2(R0)    ;WHAT IT SHOULD BE
1956  025334  104021                          ERROR   21              ;INCORRECT DCF REG. BITS
1957  025336  005060  017004          3$:     CLR     NOPRMP(R0)
1958  025342  104401  044746                  TYPE    ,TM111          ;INTERRUPT AS EXPECTED
1959  025346  106237  014712          4$:     ASRB    SYNC.1          ;WAIT FOR POWER-UP
1960  025352  103775                          BCS     4$
1961  025354  012716  024644                  MOV     #TST17, (SP)    ;CONTINUE
1962  025360  000002                          RTI
```

```
                                        ;;*****************************************************************
                                        ;*TEST 20       CHECK POWERFAIL DURING HIGH MEMORY ACTIVITY
                                        ;;*****************************************************************
025362                                  TST20:
025362  012777  000020  166600          MOV     #20,@DISPLAY    ;SET TEST NUMBER
                                        ;;***************** TS20A-FORK ****************************
025370  016006  014200                  MOV     $$STP(R0),      SP;SET UP THE STACK
025374  052777  100000  171420          BIS     #BIT15, @ACR    ;INITIALIZE THE IIST
025402  106277  000122          70$:    ASRB    @69$            ;ENTER CONTROL FORK
025406  103375                          BCC     70$
025410  027737  000110  016766          CMP     @67$,CPUACT             ;HAVE WE REACHED THE END OF THE
                                                                ;ROUTING CYCLE?
025416  001021                          BNE     65$             ;BRANCH IF NO
025420  013702  016766                  MOV     CPUACT,R2
025424  005302                          DEC     R2
025426  006302                          ASL     R2              ;(CPUACT-1)*2
025430  027702  000072                  CMP     @68$,R2         ;ARE WE AT THE END OF THE TEST?
025434  001005                          BNE     64$             ;BRANCH IF NO
025436  012777  000001  000064          MOV     #1,     @69$    ;EXIT
025444  000137  025704                  JMP     TST21
025450  062777  000002  000050  64$:    ADD     #2,     @68$            ;INCREMENT 68$ BY 2
025456  005077  000042                  CLR     @67$            ;CLEAR THE CHECKPOINT COUNTER
025462  005277  000036          65$:    INC     @67$            ;INCREMENT CHECKPOINT
025466  005037  014714                  CLR     SYNC.2          ;CLEAR THE LOCK
025472  020077  000030                  CMP     R0,@68$         ;ROUTE THIS PROCESSOR THROUGH TS20A?
025476  001005                          BNE     66$             ;BRANCH IF NO
025500  012777  000001  000022          MOV     #1,@69$         ;CLEAR LOCK
025506  000137  025532                  JMP     TS20A           ;JUMP TO BRANCH TS20A
025512  012777  000001  000010  66$:    MOV     #1,@69$         ;CLEAR LOCK
025520  000137  025634                  JMP     TS20B           ;JUMP TO TS20B
                                        ;;*************************************************************
025524  016726                  67$:    D1
025526  016730                  68$:    D2
025530  017000                  69$:    D3
025532  112761  000020  014002  TS20A:  MOVB    #20,    $TSTNM(R1)
025540  010037  017054                  MOV     R0,PUT          ;SET PROCESSOR UNDER TEST
025544  005700                          TST     R0              ;IS THIS THE MASTER?
025546  001007                          BNE     5$              ;BRANCH IF NO
025550  104401  041476                  TYPE    ,TM77           ;'TEST'
025554  005046                          CLR     -(SP)
025556  116116  014002                  MOVB    $TSTNM(R1),(SP) ;GET THE TEST NO.
025562  104403                          TYPOS
025564  000002                          .WORD   2                       ;TYPE 2 DIGITS, NO LEADING 0
025566                          5$:
025566  104401  041507                  TYPE    ,TM100          ;'POWERFAIL CPU # ''
025572  016046  014226                  MOV     $CPUID(R0),-(SP)        ;;SAVE $CPUID(R0) FOR TYPEOUT
025576  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
025600  104401  014333                  TYPE    ,$CRLF
025604  012760  014020  014612          MOV     #SSU!TI.NCX,PFFT(R0) ;SEND SIGNAL ON UP,TIME,DON'T SAVE NN
025612  005003                          CLR     R3              ;SET FOR RTI RETURN
025614  005037  014642                  CLR     SIGNAL          ;CLEAR THE POWER-UP SIGNAL
025620  012737  177777  014712          MOV     #-1,    SYNC.1  ;UNLOCK THE OTHER CPUS
025626  000001                          WAIT                    ;WAIT FOR THE POWER TO FAIL
025630  000137  025362                  JMP     TST20
```

```
 2019  025634  112761  000020  014002  TS20B:  MOVB    #20,    $TSTNM(R1);SET UP THE TEST NUMBER
 2020  025642  106237  014712                  ASRB    SYNC.1          ;WAIT FOR SYNC. SIGNAL
 2021  025646  103372                          BCC     TS20B
 2022  025650  104401  014220                  TYPE    ,$NULL          ;FLUSH THE TYPE QUEUE
 2023  025654  012705  002000          1$:     MOV     #2000,R5        ;INITIALIZE COUNTER
 2024  025660  011010                  2$:     MOV     (R0),(R0)
 2025  025662  011010                          MOV     (R0),(R0)
 2026  025664  011010                          MOV     (R0),(R0)
 2027  025666  011010                          MOV     (R0),(R0)
 2028  025670  077505                          SOB     R5,2$
 2029  025672  106237  014642                  ASRB    SIGNAL          ;SIGNAL RECEIVED?
 2030  025676  103366                          BCC     1$              ;NO,CONTINUE WITH CONTENSION
 2031  025700  000137  025362                  JMP     TST20
```

```
                                        ;;*****************************************************************
                                        ;*TEST 21       CHECK POWERFAIL SEQUENCE DURING MASSBUS XFER
                                        ;;*****************************************************************
025704                                  TST21:
025704  012777  000021  166256          MOV     #21,@DISPLAY    ;SET TEST NUMBER
                                        ;;**************** TS21A-FORK ***************************
025712  016006  014200                  MOV     $$STP(R0),      SP;SET UP THE STACK
025716  052777  100000  171076          BIS     #BIT15, @ACR    ;INITIALIZE THE LIST
025724  106277  000122          70$:    ASRB    @69$            ;ENTER CONTROL WORK
025730  103375                          BCC     70$
025732  027737  000110  016766          CMP     @67$,CPUACT             ;HAVE WE REACHED THE END OF THE
                                                                ;ROUTING CYCLE?
025740  001021                          BNE     65$             ;BRANCH IF NO
025742  013702  016766                  MOV     CPUACT,R2
025746  005302                          DEC     R2
025750  006302                          ASL     R2              ;(CPUACT-1)*2
025752  027702  000072                  CMP     @68$,R2         ;ARE WE AT THE END OF THE TEST?
025756  001005                          BNE     64$             ;BRANCH IF NO
025760  012777  000001  000064          MOV     #1,     @69$    ;EXIT
025766  000137  026632                  JMP     TST22
025772  062777  000002  000050  64$:    ADD     #2,     @68$            ;INCREMENT 68$ BY 2
026000  005077  000042                  CLR     @67$            ;CLEAR THE CHECKPOINT COUNTER
026004  005277  000036          65$:    INC     @67$            ;INCREMENT CHECKPOINT
026010  005037  014714                  CLR     SYNC.2          ;CLEAR THE LOCK
026014  020077  000030                  CMP     R0,@68$         ;ROUTE THIS PROCESSOR THROUGH TS21A?
026020  001005                          BNE     66$             ;BRANCH IF NO
026022  012777  000001  000022          MOV     #1,@69$         ;CLEAR LOCK
026030  000137  026054                  JMP     TS21A           ;JUMP TO BRANCH TS21A
026034  012777  000001  000010  66$:    MOV     #1,@69$         ;CLEAR LOCK
026042  000137  026434                  JMP     TS21B           ;JUMP TO TS21B
                                        ;;*************************************************************
026046  016732                  67$:    E1
026050  016734                  68$:    E2
026052  017002                  69$:    E3
026054  112761  000021  014002  TS21A:  MOVB    #21,    $TSTNM(R1)
026062  010037  017054                  MOV     R0,PUT          ;SET PROCESSOR UNDER TEST
026066  023737  016732  016766  4$:     CMP     E1,     CPUACT  ;LET THE OTHER CPUS CATCH UP
026074  001374                          BNE     4$
026076  005700                          TST     R0              ;IS THIS THE MASTER?
026100  001007                          BNE     5$              ;BRANCH IF NO
026102  104401  041476                  TYPE    ,TM77           ;'TEST'
026106  005046                          CLR     -(SP)
026110  116116  014002                  MOVB    $TSTNM(R1),(SP) ;GET THE TEST NO.
026114  104403                          TYPOS
026116  000002                          .WORD   2                       ;TYPE 2 DIGITS, NO LEADING 0
026120                          5$:
026120  013702  016766                  MOV     CPUACT, R2      ;CHECK FOR MASSBUS DEVICES ON OTHER CPUS
026124  006302                          ASL     R2
026126  005005                          CLR     R5
026130  020205                  1$:     CMP     R2,     R5
026132  001407                          BEQ     2$
026134  022765  001400  014632          CMP     #1400,  MBDSW(R5)
026142  001020                          BNE     3$
026144  062705  000002                  ADD     #2,     R5
026150  000767                          BR      1$
```

```
2088  026152  104401  043732          2$:     TYPE    ,NODEV          ;THERE ARE NO DEVICES TO TEST THIS CPU
2089  026156  016046  014226                  MOV     $CPUID(R0),-(SP)        ;;SAVE $CPUID(R0) FOR TYPEOUT
2090  026162  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2091  026164  104401  014333                  TYPE    ,$CRLF
2092  026170  104401  041604                  TYPE    ,TM102          ;'PROCEEDING TO NEXT CPU'
2093  026174  012737  177777  014714          MOV     #-1,    SYNC.2
2094  026202  000640                          BR      TST21           ;BRANCH TO START OF TEST
2095  026204  020005                  3$:     CMP     R0,     R5
2096  026206  001003                          BNE     P21             ;BRANCH IF THERE IS A DEVICE ON ANOTHER CPU
2097  026210  062705  000002                  ADD     #2,     R5      ;THE DEVICE IS ON THIS CPU
2098  026214  000745                          BR      1$
2099  026216  104401  041507          P21:    TYPE    ,TM100          ;'POWER FAIL CPU #'
2100  026222  016046  014226                  MOV     $CPUID(R0),-(SP)        ;;SAVE $CPUID(R0) FOR TYPEOUT
2101  026226  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2102  026230  104401  014333                  TYPE    ,$CRLF
2103  026234  012760  014020  014612          MOV     #SSU.TI!NCX,PFFT(R0) ;SEND SIGNAL ON UP TIME, DON'T SAVE NN
2104  026242  012737  177777  014712          MOV     #-1,    SYNC.1  ;UNLOCK THE OTHER CPU'S
2105  026250  005037  014642                  CLR     SIGNAL          ;CLEAR THE POWER-UP SIGNAL
2106  026254  005003                          CLR     R3              ;COME UP VIA RTI
2107  026256  022760  001400  014632          CMP     #1400,MBDSW(R0) ;DOES THIS CPU HAVE A MASSBUS DEVICE?
2108  026264  001006                          BNE     1$              ;BRANCH IF YES
2109  026266  000001                          WAIT                    ;WAIT FOR POWER TO FAIL
2110  026270  012737  177777  014714          MOV     #-1,    SYNC.2  ;UNLOCK CPUS IF ANY ARE LOCKED
2111  026276  000137  025704                  JMP     TST21           ;GO TO CONTROL FORK
2112  026302  016004  017034          1$:     MOV     BFADR(R0),R4    ;PUT ADDRESS OF BUFFER IN R4
2113  026306  012737  000070  176700          MOV     #70,@#RPCS1     ;DO A READ
2114  026314  004737  033274                  JSR     PC,MBUSR        ;READ A RECORD
2115  026320  005060  014650                  CLR     CKSUM(R0)       ;CLEAR CHECKSUM LOCATION
2116  026324  012702  004000                  MOV     #4000,R2        ;INITIALIZE A COUNTER
2117  026330  016004  017034                  MOV     BFADR(R0),R4    ;GET BUFFER POINTER
2118  026334  062460  014650          2$:     ADD     (R4)+,CKSUM(R0) ;PERFORM
2119  026340  005560  014650                  ADC     CKSUM(R0)       ;CHECKSUM
2120  026344  077205                          SOB     R2,2$           ;LOOP
2121  026346  016004  017034          4$:     MOV     BFADR(R0),R4    ;LOAD BUFFER ADDRESS
2122  026352  012737  000050  176700          MOV     #50,@#RPCS1     ;DO A WRITE CHECK
2123  026360  004737  033274                  JSR     PC,MBUSR        ;READ FROM MASS BUS DEVICE
2124
2125  026364  005005                          CLR     R5              ;CLEAR R5
2126  026366  012702  004000                  MOV     #4000,R2        ;INITIALIZE COUNTER
2127  026372  016004  017034                  MOV     BFADR(R0),R4    ;GET POINTER TO BUFFER
2128  026376  062405                  5$:     ADD     (R4)+,R5        ;PERFORM
2129  026400  005505                          ADC     R5              ;CHECKSUM
2130  026402  077203                          SOB     R2,5$           ;
2131  026404  020560  014650                  CMP     R5,CKSUM(R0)    ;EVERYTHING OK?
2132  026410  001401                          BEQ     6$              ;BRANCH IF YES
2133  026412  104023                          ERROR   23              ;CHECKSUM IS WRONG
2134  026414  106237  014642          6$:     ASRB    SIGNAL          ;
2135  026420  103352                          BCC     4$              ;NO CONTINUE XFERS
2136  026422  012737  177777  014714          MOV     #-1,SYNC.2      ;UNLICK CPUS IF ANY ARE LOCKED
2137  026430  000137  025704                  JMP     TST21           ;GO TO CONTROL FORK
2138  026434  112761  000021  014002  TS21B:  MOVB    #21,    $TSTNM(R1)      ;SET THE TEST NUMBER
2139  026442  022760  001400  014632          CMP     #1400,  MBDSW(R0)       ;DOES THIS CPU HAVE A MASSBUS DEVICE?
2140  026450  001007                          BNE     1$              ;BRANCH IF YES
2141  026452  106237  014714          10$:    ASRB    SYNC.2          ;ELSE SET OUT THIS ROUND
2142  026456  103375                          BCC     10$
2143  026460  104401  014220                  TYPE    ,$NULL          ;FLUSH THE TYPE QUEUE
```

```
2144  026464  000137  025704                  JMP     TST21           ;JUMP INTO THE CONTROL LOOP
2145  026470                          1$:
2146  026470  016004  017034                  MOV     BFADR(R0),R4    ;PUT ADDRESS OF BUFFER IN R4
2147  026474  012737  000070  176700          MOV     #70,@#RPCS1     ;DO A READ
2148  026502  004737  033274                  JSR     PC,MBUSR        ;READ A RECORD
2149  026506  005060  014650                  CLR     CKSUM(R0)               ;CLEAR CHECKSUM LOCATION
2150  026512  012703  004000                  MOV     #4000,R3                ;INITIALIZE A COUNTER
2151  026516  016004  017034                  MOV     BFADR(R0),R4    ;GET BUFFER POINTER
2152  026522  062460  014650          2$:     ADD     (R4)+,CKSUM(R0) ;PERFORM...
2153  026526  005560  014650                  ADC     CKSUM(R0)               ;CHECKSUM.
2154  026532  077305                          SOB     R3,2$           ;LOOP
2155  026534  106237  014712          7$:     ASRB    SYNC.1          ;HOLD UP
2156  026540  103375                          BCC     7$
2157  026542  104401  014220                  TYPE    ,$NULL          ;FLUSH THE QUEUE
2158  026546  016004  017034          4$:     MOV     BFADR(R0),R4    ;LOAD BUFFER ADDRESS
2159  026552  012737  000050  176700          MOV     #50,@#RPCS1     ;DO A WRITE CHECK
2160  026560  004737  033274                  JSR     PC,MBUSR        ;READ FROM MASS BUS DEVICE
2161  026564  005005                          CLR     R5              ;CLEAR R5
2162  026566  012703  004000                  MOV     #4000,R3                ;INITIALIZE COUNTER
2163  026572  016004  017034                  MOV     BFADR(R0),R4    ;GET POINTER TO BUFFER
2164  026576  062405                  5$:     ADD     (R4)+,R5        ;PERFORM...
2165  026600  005505                          ADC     R5              ;CHECKSUM
2166  026602  077303                          SOB     R3,5$
2167  026604  020560  014650                  CMP     R5,CKSUM(R0)    ;EVERYTHING OK?
2168  026610  001401                          BEQ     6$              ;BRANCH IF YES
2169  026612  104023                          ERROR   23              ;CHECKSUM IS WRONG
2170  026614  106237  014642          6$:     ASRB    SIGNAL
2171  026620  103352                          BCC     4$              ;NO CONTINUE XFERS
2172  026622  104401  014220                  TYPE    ,$NULL
2173  026626  000137  025704                  JMP     TST21           ;JUMP INTO CONTROI LOOP
2174
2175
2176
2177
```

```
                                        ;;*********************************************************************
                                        ;*TEST 22       CHECK AC POWERFAIL ON MEM BOXES, PORTS DISABLED
                                        ;;*********************************************************************
026632                                  TST22:
026632  012777  000022  165330          MOV     #22,@DISPLAY    ;SET TEST NUMBER

026640  112761  000022  014002          MOVB    #22,    $TSTNM(R1)      ;SET THE TEST NUMBER
026646  106237  016770          2$:     ASRB    SYNC.3                  ;CONTROL THE ENTRY
026652  103375                          BCC     2$
026654  005237  014542                  INC     ENTR22                  ;INCREMENT ENTER FLAG
026660  012737  000001  016770          MOV     #1,     SYNC.3          ;ALLOW THE OTHERS IN
026666  023737  016766  014542  1$:     CMP     CPUACT, ENTR22          ;ARE ALL CPUS HERE?
026674  001374                          BNE     1$                      ;NOT YET
026676  005037  014540                  CLR     EXIT                    ;CLEAR THE EXIT FLAG
026702  005037  177776                  CLR     PSW                     ;SET KERNAL MODE
026706  C20027  000000                  CMP     R0,     #0              ;IS THIS THE MASTER?
026712  001001                          BNE     TS22B                   ;BRANCH IF NO
026714  000474                          BR      TS22A                   ;THIS IS THE MASTER

026716  052777  100000  170076  TS22B:  BIS     #BIT15, @ACR            ;INITIALIZE THE IIST
026724  012777  000022  165236          MOV     #22,    @DISPLAY        ;SET TEST NUMBER
026732  016006  014200                  MOV     $$STP(R0),      SP      ;INITIALIZE THE STACK
026736  005060  014612                  CLR     PFFT(R0)                ;SPECIFY THE POWER FAIL
026742  012760  040362  014702          MOV     #CPUER, ERRTAB(R0)      ;SET FOR UNEXPECTED TRAPS TO 4
026750  012703  027062                  MOV     #100$,  R3              ;SET FOR POWER FAIL RETURN
026754  005737  014540                  TST     EXIT                    ;FINISHED WITH THIS TEST?
026760  001404                          BEQ     1$                      ;BRANCH IF NO
026762  005037  177572                  CLR     MMR0                    ;MAKE SURE MM IS TURNED OFF
026766  000137  027654                  JMP     TST23                   ;GO TO NEXT TEST
026772  005737  014534          1$:     TST     RELOUP                  ;TIME TO RELOCATE?
026776  001421                          BEQ     2$                      ;BRANCH IF NO
027000  004737  032340                  JSR     PC,     SETMM           ;GET READY FOR RELOCATION
027004  063737  014550  172340          ADD     HIBOX,  KIPAR0
027012  063737  014550  172342          ADD     HIBOX,  KIPAR1
027020  063737  014550  172344          ADD     HIBOX,  KIPAR2
027026  052737  000001  177572          BIS     #1,     MMR0            ;SLAVE IS NOW IN HIGH CORE
027034  005037  014534                  CLR     RELOUP                  ;CLEAR RELOCATION FLAG
027040  000726                          BR      TS22B                   ;CONTINUE TESTING
027042  005737  014536          2$:     TST     RELODN                  ;TIME TO RELOCATE?
027046  001723                          BEQ     TS22B                   ;BRANCH IF NO
027050  005037  177572                  CLR     MMR0                    ;RETURN TO LOW CORE
027054  005037  014536                  CLR     RELODN                  ;CLEAR THE FLAG
027060  000716                          BR      TS22B                   ;CONTINUE

027062  005737  014522          100$:   TST     PWRFL                   ;SHOULD WE BE HERE?
027066  001002                          BNE     101$                    ;BRANCH IF YES
027070  104001                          ERROR   1                       ;UNEXPECTED CPU POWER FAIL
027072  000711                          BR      TS22B                   ;CONTINUE TESTING
027074  005060  017004          101$:   CLR     NOPRMP(R0)              ;WANT TO IDENTIFY THE CPU
027100  104401  041635                  TYPE    ,TM103                  ;EXPECTED CPU POWER FAI_
027104  000704                          BR      TS22B                   ;CONTINUE TESTING


027106                          TS22A:
027106  005700                          TST     R0              ;IS THIS THE MASTER?
```

```
2234  027110  001007                          BNE     5$                  ;BRANCH IF NO
2235  027112  104401  041476                  TYPE    ,TM77               ;"TEST"
2236  027116  005046                          CLR     -(SP)
2237  027120  116116  014002                  MOVB    $TSTNM(R1),(SP) ;GET THE TEST NO.
2238  027124  104403                          TYPOS
2239  027126  000002                          .WORD   2                           ;TYPE 2 DIGITS, NO LEADING 0
2240  027130                          5$:
2241  027130  012737  000000  014522          MOV     #0,     PWRFL               ;SPECIFY WHETHER OR NOT TO EXPECT CPU POWER FAIL
2242  027136  012737  000000  014526          MOV     #0,     BOOT                ;SPECIFY WHETHER OR NOT TO EXPECT CPU BOOT AND I
2243  027144  004737  032076                  JSR     PC,     MEMSIZ              ;FIND ALL THE MEM BOXES
2244  027150  005737  014540                  TST     EXIT                        ;WAS ONLY ONE MEM BOX FOUND?
2245  027154  001402                          BEQ     1$                          ;BRANCH IF NO
2246  027156  000137  027654                  JMP     TST23                       ;WE CAN'T DO THIS TEST
2247  027162  012702  000016          1$:     MOV     #16,    R2                  ;POINT TO BOX #7
2248  027166  052777  100000  167626  2$:     BIS     #BIT15, @ACR                ;INITIALIZE THE IIST
2249  027174  012777  000022  164766          MOV     #22,    @DISPLAY            ;SET THE TEST NUMBER
2250  027202  016006  014200                  MOV     $$STP(R0),      SP          ;INIALIZE THE STACK
2251  027206  005060  014612                  CLR     PFFT(R0)                    ;SPECIFY THE POWER FAIL
2252  027212  012760  040362  014702          MOV     #CPUER, ERRTAB(R0)          ;SET FOR UNEXPECTED TRAPS TO 4
2253  027220  005702                          TST     R2                          ;DID WE TEST ALL THE BOXES
2254  027222  002012                          BGE     3$                          ;BRANCH IF NO
2255  027224  012737  000112  000110          MOV     #112,   @#110               ;RESTORE LOC 110
2256  027232  005037  000000                  CLR     @#0                         ;RESTORE LOC 0
2257  027236  012737  000001  014540          MOV     #1,     EXIT                ;SIGNAL THE SLAVES TO EXIT
2258  027244  000137  027654                  JMP     TST23                       ;GO TO THE NEXT TEST
2259  027250  005762  014502          3$:     TST     START(R2)                   ;WHAT DO WE KNOW ABOUT THIS BOX?
2260  027254  003037                          BGT     10$                         ;BRANCH IF NOT THE BASE BOX
2261  027256  001403                          BEQ     4$                          ;BRANCH IF ITS THE BASE BOX
2262  027260  162702  000002                  SUB     #2,     R2                  ;THERE WAS NO BOX-POINT TO THE NEXT LOWER BOX
2263  027264  000740                          BR      2$                          ;CONTINUE
2264  027266  022737  000001  014646  4$:     CMP     #1,     BOXNUM              ;WAS THERE ONLY ONE MEM BOX?
2265  027274  002027                          BGE     11$                         ;BRANCH IF YES
2266  027276  005737  014526                  TST     BOOT                        ;IS THIS THE DC TEST?
2267  027302  001141                          BNE     103$                        ;BRANCH IF YES
2268  027304  004737  032576                  JSR     PC,     RELOHI              ;TO TEST THE BASE BOX,
2269                                                                              ;RELOCATE THE PROGRAM FIRST TO THE NEXT HIGHER B
2270  027310  012737  000001  014532          MOV     #1,     HICORE              ;TURN MM ON ON POWER-UP
2271  027316  012737  000001  014534          MOV     #1,     RELOUP              ;SIGNAL SLAVES TO RELOCATE
2272  027324  063737  014550  172340          ADD     HIBOX,  KIPAR0              ;GET READY TO GO TO HIGH CORE
2273  027332  063737  014550  172342          ADD     HIBOX,  KIPAR1
2274  027340  063737  014550  172344          ADD     HIBOX,  KIPAR2
2275  027346  052737  000001  177572          BIS     #1,     MMR0                ;WE ARE NOW IN HIGH CORE
2276  027354                          10$:
2277  027354  005037  014530          11$.    CLR     PATCHK                      ;SET TO WRITE PATTERN
2278  027360  004737  032434                  JSR     PC,     PATTRN              ;WRITE THE PATTERN
2279  027364  004737  032322                  JSR     PC,     BUFCLR              ;CLEAR THE KEYBOARD BUFFER
2280  027370  104401  042317                  TYPE    ,TM106                      ;TELL THE OPERATOR TO POWER FAIL THE MEMORY BOX
2281  027374  006202                          ASR     R2
2282  027376  010246                          MOV     R2,     -(SP)
2283  027400  104405                          TYPDS
2284  027402  006302                          ASL     R2
2285  027404  104401  043011                  TYPE    ,TM108                      ;SPECIFY THE CONDITIONS
2286  027410  012703  027552                  MOV     #100$,  R3                  ;SET UP THE POWER FAIL RETURN
2287  027414  005737  014526                  TST     BOOT                        ;IS THIS THE DC TEST?
2288  027420  001414                          BEQ     12$                         ;BRANCH IF NO
2289  027422  012737  020000  000000  13$:     MOV     #20000, @#0                 ;MAKE BOOTING SLAVE SP=20000
```

```
2290  027430  005237  000110              INC     @#110                   ;HANG THE SLAVES BOOT
2291  027434  105777  164550              TSTB    @$TKS                   ;IS CHARACTER IN BUFFER?
2292  027440  100370                      BPL     13$                     ;LOOP
2293  027442  012777  032762  167356      MOV     #ENTR,  @ISTVEC         ;SET UP TO INTERRUPT SLAVES
2294  027450  000413                      BR      14$                     ;CONTINUE
2295  027452                      12$:
2296  027452  105777  164532              TSTB    @$TKS                   ;IS CHARACTER IN THE BUFFER?
2297  027456  100375                      BPL     12$                     ;LOOP
2298  027460  005737  014522      17$:    TST     PWRFL                   ;SHOULD THE MASTER HAVE POWER FAILED?
2299  027464  001402                      BEQ     18$                     ;BRANCH IF NO
2300  027466  104024                      ERROR   24                      ;FAILURE TO POWER FAIL
2301  027470  000441                      BR      102$                    ;CONTINUE
2302  027472  104401  047036      18$:    TYPE    ,OK
2303  027476  000436                      BR      102$
2304  027500  017704  164454      14$:    MOV     @SWR,   R4              ;GET THE SWITCH VALUES
2305  027504  042704  177760              BIC     #177760,        R4      ;SAVE ONLY CPU BITS
2306  027510  012777  000000  167304      MOV     #PGTE,  @ACR            ;ACCESS PGTE REG
2307  027516  010477  167302              MOV     R4,     @ADR            ;SET INTERRUPT BITS
2308  027522  032737  000001  016766      BIT     #BIT0,  CPUACT          ;EVEN OR ODD?
2309  027530  001404                      BEQ     15$                     ;BRANCH IF EVEN
2310  027532  012777  000001  167264      MOV     #1,     @ADR            ;INTERRUPTING AN EVEN NUMBER OF SLAVES
2311  027540  000422                      BR      103$                    ;
2312  027542  012777  000003  167254  15$:MOV     #3,     @ADR            ;INTERRUPTING AN ODD # OF SLAVES
2313  027550  000416                      BR      103$
2314
2315  027552  005737  014522      100$:   TST     PWRFL                   ;DID WE EXPECT POWER FAIL?
2316  027556  001002                      BNE     101$                    ;BRANCH IF YES
2317  027560  104001                      ERROR   1                       ;MASTER ERROREOUSLY POWER FAILED
2318  027562  000404                      BR      102$                    ;CONTINUE
2319  027564  005060  017004      101$:   CLR     NOPRMP(R0)              ;ALLOW CPU IDENTIFICATION
2320  027570  104401  041635              TYPE    ,TM103                  ;CPU POWER FAIL MSG
2321  027574  012737  000001  014530  102$:MOV     #1,     PATCHK          ;SET UP FOR PATTERN CHECK
2322  027602  004737  032434              JSR     PC,     PATTRN          ;CHECK THE PATTERN
2323  027606  032737  000001  177572  103$:BIT     #1,     MMR0            ;ARE WE IN HIGH CORE?
2324  027614  001411                      BEQ     104$                    ;BRANCH IF NO
2325  027616  004737  032722              JSR     PC,     RELOLO          ;ELSE RELOCATE
2326  027622  012737  000001  014536      MOV     #1,     RELODN          ;SIGNAL THE SLAVES
2327  027630  005037  177572              CLR     MMR0                    ;WE ARE NOW BACK DOWN IN LOW CORE
2328  027634  005037  014532              CLR     HICORE                  ;MAKE SURE MM ON POWER-UP DISABLED
2329  027640  005005                      CLR     R5                      ;MAKE TS22A DELAY
2330  027642  077501                      SOB     R5,     .
2331  027644  162702  000002              SUB     #2,     R2              ;POINT TO NEXT BOX
2332  027650  000137  027166              JMP     2$                      ;CONTINUE
2333
2334
2335
```

```
                                ;;*****************************************************************
                                ;*TEST 23       CHECK AC POWERFAIL ON MEM BOXES, PORTS ENABLED
                                ;;*****************************************************************
027654                          TST23:
027654  012777  000023  164306          MOV     #23,@DISPLAY    ;SET TEST NUMBER

027662  112761  000023  014002          MOVB    #23,    $TSTNM(R1)      ;SET THE TEST NUMBER
027670  106237  016770          2$:     ASRB    SYNC.3                  ;CONTROL THE ENTRY
027674  103375                          BCC     2$
027676  005237  014544                  INC     ENTR23                  ;INCREMENT ENTER FLAG
027702  012737  000001  016770          MOV     #1,     SYNC.3          ;ALLOW THE OTHERS IN
027710  023737  016766  014544  1$:     CMP     CPUACT, ENTR23          ;ARE ALL CPUS HERE?
027716  001374                          BNE     1$                      ;NOT YET
027720  005037  014540                  CLR     EXIT                    ;CLEAR THE EXIT FLAG
027724  005037  177776                  CLR     PSW                     ;SET KERNAL MODE
027730  020027  000000                  CMP     R0,     #0              ;IS THIS THE MASTER?
027734  001001                          BNE     TS23B                   ;BRANCH IF NO
027736  000474                          BR      TS23A                   ;THIS IS THE MASTER

027740  052777  100000  16/054  TS23B:  BIS     #BIT15, @ACR            ;INITIALIZE THE IIST
027746  012777  000023  164214          MOV     #23,    @DISPLAY        ;SET TEST NUMBER
027754  016006  014200                  MOV     $$STP(R0),      SP      ;INITIALIZE THE STACK
027760  005060  014612                  CLR     PFFT(R0)                ;SPECIFY THE POWER FAIL
027764  012760  040362  014702          MOV     #CPUER, ERRTAB(R0)      ;SET FOR UNEXPECTED TRAPS TO 4
027772  012703  030104                  MOV     #100$,  R3              ;SET FOR POWER FAIL RETURN
027776  005737  014540                  TST     EXIT                    ;FINISHED WITH THIS TEST?
030002  001404                          BEQ     1$                      ;BRANCH IF NO
030004  005037  177572                  CLR     MMR0                    ;MAKE SURE MM IS TURNED OFF
030010  000137  030676                  JMP     TST24                   ;GO TO NEXT TEST
030014  005737  014534          1$:     TST     RELOUP                  ;TIME TO RELOCATE?
030020  001421                          BEQ     2$                      ;BRANCH IF NO
030022  004737  032340                  JSR     PC,     SETMM           ;GET READY FOR RELOCATION
030026  063737  014550  172340          ADD     HIBOX,  KIPAR0
030034  063737  014550  172342          ADD     HIBOX,  KIPAR1
030042  063737  014550  172344          ADD     HIBOX,  KIPAR2
030050  052737  000001  177572          BIS     #1,     MMR0            ;SLAVE IS NOW IN HIGH CORE
030056  005037  014534                  CLR     RELOUP                  ;CLEAR RELOCATION FLAG
030062  000726                          BR      TS23B                   ;CONTINUE TESTING
030064  005737  014536          2$:     TST     RELODN                  ;TIME TO RELOCATE?
030070  001723                          BEQ     TS23B                   ;BRANCH IF NO
030072  005037  177572                  CLR     MMR0                    ;RETURN TO LOW CORE
030076  005037  014536                  CLR     RELODN                  ;CLEAR THE FLAG
030102  000716                          BR      TS23B                   ;CONTINUE

030104  005737  014522          100$:   TST     PWRFL                   ;SHOULD WE BE HERE?
030110  001002                          BNE     101$                    ;BRANCH IF YES
030112  104001                          ERROR   1                       ;UNEXPECTED CPU POWER FAIL
030114  000711                          BR      TS23B                   ;CONTINUE TESTING
030116  005060  017004          101$:   CLR     NOPRMP(R0)              ;WANT TO IDENTIFY THE CPU
030122  104401  041635                  TYPE    ,TM103                  ;EXPECTED CPU POWER FAIL
030126  000704                          BR      TS23B                   ;CONTINUE TESTING


030130                          TS23A:
030130  005700                          TST     R0              ;IS THIS THE MASTER?
```

```
2392  030132  001007                          BNE     5$              ;BRANCH IF NO
2393  030134  104401  041476                  TYPE    ,TM77           ;"TEST"
2394  030140  005046                          CLR     -(SP)
2395  030142  116116  014002                  MOVB    $TSTNM(R1),(SP) ;GET THE TEST NO.
2396  030146  104403                          TYPOS
2397  030150  000002                          .WORD   2                       ;TYPE 2 DIGITS, NO LEADING 0
2398  030152                          5$:
2399  030152  012737  000001  014522          MOV     #1,     PWRFL           ;SPECIFY WHETHER OR NOT TO EXPECT CPU POWER FAIL
2400  030160  012737  000000  014526          MOV     #0,     BOOT            ;SPECIFY WHETHER OR NOT TO EXPECT CPU BOOT AND I
2401  030166  004737  032076                  JSR     PC,     MEMSIZ          ;FIND ALL THE MEM BOXES
2402  030172  005737  014540                  TST     EXIT                    ;WAS ONLY ONE MEM BOX FOUND?
2403  030176  001402                          BEQ     1$                      ;BRANCH IF NO
2404  030200  000137  030676                  JMP     TST24                   ;WE CAN'T DO THIS TEST
2405  030204  012702  000016          1$:     MOV     #16,    R2              ;POINT TO BOX #7
2406  030210  052777  100000  166604  2$:     BIS     #BIT15, @ACR            ;INITIALIZE THE IIST
2407  030216  012777  000023  163744          MOV     #23,    @DISPLAY        ;SET THE TEST NUMBER
2408  030224  016006  014200                  MOV     $$STP(R0),      SP      ;INIALIZE THE STACK
2409  030230  005060  014612                  CLR     PFFT(R0)                ;SPECIFY THE POWER FAIL
2410  030234  012760  040362  014702          MOV     #CPUER, ERRTAB(R0)      ;SET FOR UNEXPECTED TRAPS TO 4
2411  030242  005702                          TST     R2                      ;DID WE TEST ALL THE BOXES
2412  030244  002012                          BGE     3$                      ;BRANCH IF NO
2413  030246  012737  000112  000110          MOV     #112,   @#110           ;RESTORE LOC 110
2414  030254  005037  000000                  CLR     @#0                     ;RESTORE LOC 0
2415  030260  012737  000001  014540          MOV     #1,     EXIT            ;SIGNAL THE SLAVES TO EXIT
2416  030266  000137  030676                  JMP     TST24                   ;GO TO THE NEXT TEST
2417  030272  005762  014502          3$:     TST     START(R2)               ;WHAT DO WE KNOW ABOUT THIS BOX?
2418  030276  003037                          BGT     10$                     ;BRANCH IF NOT THE BASE BOX
2419  030300  001403                          BEQ     4$                      ;BRANCH IF ITS THE BASE BOX
2420  030302  162702  000002                  SUB     #2,     R2              ;THERE WAS NO BOX-POINT TO THE NEXT LOWER BOX
2421  030306  000740                          BR      2$                      ;CONTINUE
2422  030310  022737  000001  014646  4$:     CMP     #1,     BOXNUM          ;WAS THERE ONLY ONE MEM BOX?
2423  030316  002027                          BGE     11$                     ;BRANCH IF YES
2424  030320  005737  014526                  TST     BOOT                    ;IS THIS THE DC TEST?
2425  030324  001141                          BNE     103$                    ;BRANCH IF YES
2426  030326  0C4737  032576                  JSR     PC,     RELOHI          ;TO TEST THE BASE BOX,
2427                                                                          ;RELOCATE THE PROGRAM FIRST TO THE NEXT HIGHER B
2428  030332  012737  000001  014532          MOV     #1,     HICORE          ;TURN MM ON ON POWER-UP
2429  030340  012737  000001  014534          MOV     #1,     RELOUP          ;SIGNAL SLAVES TO RELOCATE
2430  030346  063737  014550  172340          ADD     HIBOX,  KIPAR0          ;GET READY TO GO TO HIGH CORE
2431  030354  063737  014550  172342          ADD     HIBOX,  KIPAR1
2432  030362  063737  014550  172344          ADD     HIBOX,  KIPAR2
2433  030370  052737  000001  177572          BIS     #1,     MMR0            ;WE ARE NOW IN HIGH CORE
2434  030376                          10$:
2435  030376  005037  014530          11$:    CLR     PATCHK                  ;SET TO WRITE PATTERN
2436  030402  004737  032434                  JSR     PC,     PATTRN          ;WRITE THE PATTERN
2437  030406  004737  032322                  JSR     PC,     BUFCLR          ;CLEAR THE KEYBOARD BUFFER
2438  030412  104401  042317                  TYPE    ,TM106                  ;TELL THE OPERATOR TO POWER FAIL THE MEMORY BOX
2439  030416  006202                          ASR     R2
2440  030420  010246                          MOV     R2,     -(SP)
2441  030422  104405                          TYPDS
2442  030424  006302                          ASL     R2
2443  030426  104401  042363                  TYPE    ,TM107                  ;SPECIFY THE CONDITIONS
2444  030432  012703  030574                  MOV     #100$,  R3              ;SET UP THE POWER FAIL RETURN
2445  030436  005737  014526                  TST     BOOT                    ;IS THIS THE DC TEST?
2446  030442  001414                          BEQ     12$                     ;BRANCH IF NO
2447  030444  012737  020000  000000  13$:    MOV     #20000, @#0             ;MAKE BOOTING SLAVE SP=20000
```

```
2448  030452  005237  000110              INC     @#110                   ;HANG THE SLAVES BOOT
2449  030456  105777  163526              TSTB    @$TKS                   ;IS CHARACTER IN BUFFER?
2450  030462  100370                      BPL     13$                     ;LOOP
2451  030464  012777  032762  166334      MOV     #ENTR,  @ISTVEC         ;SET UP TO INTERRUPT SLAVES
2452  030472  000413                      BR      14$                     ;CONTINUE
2453  030474                      12$:
2454  030474  105777  163510              TSTB    @$TKS                   ;IS CHARACTER IN THE BUFFER?
2455  030500  100375                      BPL     12$                     ;LOOP
2456  030502  005737  014522      17$:    TST     PWRFL                   ;SHOULD THE MASTER HAVE POWER FAILED?
2457  030506  001402                      BEQ     18$                     ;BRANCH IF NO
2458  030510  104024                      ERROR   24                      ;FAILURE TO POWER FAIL
2459  030512  000441                      BR      102$                    ;CONTINUE
2460  030514  104401  047036      18$:    TYPE    ,OK
2461  030520  000436                      BR      102$
2462  030522  017704  163432      14$:    MOV     @SWR,   R4              ;GET THE SWITCH VALUES
2463  030526  042704  177760              BIC     #177760,        R4      ;SAVE ONLY CPU BITS
2464  030532  012777  000000  166262      MOV     #PGTE,  @ACR            ;ACCESS PGTE REG
2465  030540  010477  166260              MOV     R4,     @ADR            ;SET INTERRUPT BITS
2466  030544  032737  000001  016766      BIT     #BIT0,  CPUACT          ;EVEN OR ODD?
2467  030552  001404                      BEQ     15$                     ;BRANCH IF EVEN
2468  030554  012777  000001  166242      MOV     #1,     @ADR            ;INTERRUPTING AN EVEN NUMBER OF SLAVES
2469  030562  000422                      BR      103$                    ;
2470  030564  012777  000003  166232  15$:MOV     #3,     @ADR            ;INTERRUPTING AN ODD # OF SLAVES
2471  030572  000416                      BR      103$
2472
2473  030574  005737  014522      100$:   TST     PWRFL                   ;DID WE EXPECT POWER FAIL?
2474  030600  001002                      BNE     101$                    ;BRANCH IF YES
2475  030602  104001                      ERROR   1                       ;MASTER ERROREOUSLY POWER FAILED
2476  030604  000404                      BR      102$                    ;CONTINUE
2477  030606  005060  017004      101$:   CLR     NOPRMP(R0)              ;ALLOW CPU IDENTIFICATION
2478  030612  104401  041635              TYPE    ,TM103                  ;CPU POWER FAIL MSG
2479  030616  012737  000001  014530  102$:MOV     #1,     PATCHK          ;SET UP FOR PATTERN CHECK
2480  030624  004737  032434              JSR     PC,     PATTRN          ;CHECK THE PATTERN
2481  030630  032737  000001  177572  103$:BIT     #1,     MMR0            ;ARE WE IN HIGH CORE?
2482  030636  001411                      BEQ     104$                    ;BRANCH IF NO
2483  030640  004737  032722              JSR     PC,     RELOLO          ;ELSE RELOCATE
2484  030644  012737  000001  014536      MOV     #1,     RELODN          ;SIGNAL THE SLAVES
2485  030652  005037  177572              CLR     MMR0                    ;WE ARE NOW BACK DOWN IN LOW CORE
2486  030656  005037  014532              CLR     HICORE                  ;MAKE SURE MM ON POWER-UP DISABLED
2487  030662  005005                      CLR     R5                      ;MAKE TS23A DELAY
2488  030664  077501                      SOB     R5,     .
2489  030666  162702  000002              SUB     #2,     R2              ;POINT TO NEXT BOX
2490  030672  000137  030210              JMP     2$                      ;CONTINUE
2491
2492
```

```
                                        ;;*****************************************************************
                                        ;*TEST 24       CHECK DC POWERFAIL ON MEM BOXES, CPUS BOOT ON POWER UP
                                        ;;*****************************************************************
030676                                  TST24:
030676  012777  000024  163264          MOV     #24,@DISPLAY    ;SET TEST NUMBER

030704  112761  000024  014002          MOVB    #24,    $TSTNM(R1)      ;SET THE TEST NUMBER
030712  106237  016770          2$:     ASRB    SYNC.3                  ;CONTROL THE ENTRY
030716  103375                          BCC     2$
030720  005237  014546                  INC     ENTR24                  ;INCREMENT ENTER FLAG
030724  012737  000001  016770          MOV     #1,     SYNC.3          ;ALLOW THE OTHERS IN
030732  023737  016766  014546  1$:     CMP     CPUACT, ENTR24          ;ARE ALL CPUS HERE?
030740  001374                          BNE     1$                      ;NOT YET
030742  005037  014540                  CLR     EXIT                    ;CLEAR THE EXIT FLAG
030746  005037  177776                  CLR     PSW                     ;SET KERNAL MODE
030752  020027  000000                  CMP     R0,     #0              ;IS THIS THE MASTER?
030756  001001                          BNE     TS24B                   ;BRANCH IF NO
030760  000474                          BR      TS24A                   ;THIS IS THE MASTER

030762  052777  100000  166032  TS24B:  BIS     #BIT15, @ACR            ;INITIALIZE THE IIST
030770  012777  000024  163172          MOV     #24,    @DISPLAY        ;SET TEST NUMBER
030776  016006  014200                  MOV     $$STP(R0),      SP      ;INITIALIZE THE STACK
031002  005060  014612                  CLR     PFFT(R0)                ;SPECIFY THE POWER FAIL
031006  012760  040362  014702          MOV     #CPUER, ERRTAB(R0)      ;SET FOR UNEXPECTED TRAPS TO 4
031014  012703  031126                  MOV     #100$,  R3              ;SET FOR POWER FAIL RETURN
031020  005737  014540                  TST     EXIT                    ;FINISHED WITH THIS TEST?
031024  001404                          BEQ     1$                      ;BRANCH IF NO
031026  005037  177572                  CLR     MMR0                    ;MAKE SURE MM IS TURNED OFF
031032  000137  031720                  JMP     TST25                   ;GO TO NEXT TEST
031036  005737  014534          1$:     TST     RELOUP                  ;TIME TO RELOCATE?
031042  001421                          BEQ     2$                      ;BRANCH IF NO
031044  004737  032340                  JSR     PC,     SETMM           ;GET READY FOR RELOCATION
031050  063737  014550  172340          ADD     HIBOX,  KIPAR0
031056  063737  014550  172342          ADD     HIBOX,  KIPAR1
031064  063737  014550  172344          ADD     HIBOX,  KIPAR2
031072  052737  000001  177572          BIS     #1,     MMR0            ;SLAVE IS NOW IN HIGH CORE
031100  005037  014534                  CLR     RELOUP                  ;CLEAR RELOCATION FLAG
031104  000726                          BR      TS24B                   ;CONTINUE TESTING
031106  005737  014536          2$:     TST     RELODN                  ;TIME TO RELOCATE?
031112  001723                          BEQ     TS24B                   ;BRANCH IF NO
031114  005037  177572                  CLR     MMR0                    ;RETURN TO LOW CORE
031120  005037  014536                  CLR     RELODN                  ;CLEAR THE FLAG
031124  000716                          BR      TS24B                   ;CONTINUE

031126  005737  014522          100$:   TST     PWRFL                   ;SHOULD WE BE HERE?
031132  001002                          BNE     101$                    ;BRANCH IF YES
031134  104001                          ERROR   1                       ;UNEXPECTED CPU POWER FAIL
031136  000711                          BR      TS24B                   ;CONTINUE TESTING
031140  005060  017004          101$:   CLR     NOPRMP(R0)              ;WANT TO IDENTIFY THE CPU
031144  104401  041635                  TYPE    ,TM103                  ;EXPECTED CPU POWER FAIL
031150  000704                          BR      TS24B                   ;CONTINUE TESTING


031152                          TS24A:
031152  005700                          TST     R0              ;IS THIS THE MASTER?
```

```
2549  031154  001007                          BNE     5$              ;BRANCH IF NO
2550  031156  104401  041476                  TYPE    ,TM77           ;"TEST"
2551  031162  005046                          CLR     -(SP)
2552  031164  116116  014002                  MOVB    $TSTNM(R1),(SP) ;GET THE TEST NO.
2553  031170  104403                          TYPOS
2554  031172  000002                          .WORD   2               ;TYPE 2 DIGITS, NO LEADING 0
2555  031174                          5$:
2556  031174  012737  000000  014522          MOV     #0,     PWRFL   ;SPECIFY WHETHER OR NOT TO EXPECT CPU POWER FAIL
2557  031202  012737  000001  014526          MOV     #1,     BOOT    ;SPECIFY WHETHER OR NOT TO EXPECT CPU BOOT AND I
2558  031210  004737  032076                  JSR     PC,     MEMSIZ  ;FIND ALL THE MEM BOXES
2559  031214  005737  014540                  TST     EXIT            ;WAS ONLY ONE MEM BOX FOUND?
2560  031220  001402                          BEQ     1$              ;BRANCH IF NO
2561  031222  000137  031720                  JMP     TST25           ;WE CAN'T DO THIS TEST
2562  031226  012702  000016          1$:     MOV     #16,    R2      ;POINT TO BOX #7
2563  031232  052777  100000  165562  2$:     BIS     #BIT15, @ACR    ;INITIALIZE THE LIST
2564  031240  012777  000024  162722          MOV     #24,    @DISPLAY ;SET THE TEST NUMBER
2565  031246  016006  014200                  MOV     $$STP(R0),      SP      ;INIALIZE THE STACK
2566  031252  005060  014612                  CLR     PFFT(R0)        ;SPECIFY THE POWER FAIL
2567  031256  012760  040362  014702          MOV     #CPUER, ERRTAB(R0)      ;SET FOR UNEXPECTED TRAPS TO 4
2568  031264  005702                          TST     R2              ;DID WE TEST ALL THE BOXES
2569  031266  002012                          BGE     3$              ;BRANCH IF NO
2570  031270  012737  000112  000110          MOV     #112,   @#110   ;RESTORE LOC 110
2571  031276  005037  000000                  CLR     @#0             ;RESTORE LOC 0
2572  031302  012737  000001  014540          MOV     #1,     EXIT    ;SIGNAL THE SLAVES TO EXIT
2573  031310  000137  031720                  JMP     TST25           ;GO TO THE NEXT TEST
2574  031314  005762  014502          3$:     TST     START(R2)       ;WHAT DO WE KNOW ABOUT THIS BOX?
2575  031320  003037                          BGT     10$             ;BRANCH IF NOT THE BASE BOX
2576  031322  001403                          BEQ     4$              ;BRANCH IF ITS THE BASE BOX
2577  031324  162702  000002                  SUB     #2,     R2      ;THERE WAS NO BOX-POINT TO THE NEXT LOWER BOX
2578  031330  000740                          BR      2$              ;CONTINUE
2579  031332  022737  000001  014646  4$:     CMP     #1,     BOXNUM  ;WAS THERE ONLY ONE MEM BOX?
2580  031340  002027                          BGE     11$             ;BRANCH IF YES
2581  031342  005737  014526                  TST     BOOT            ;IS THIS THE DC TEST?
2582  031346  001141                          BNE     103$            ;BRANCH IF YES
2583  031350  004737  032576                  JSR     PC,     RELOHI  ;TO TEST THE BASE BOX,
2584                                                                  ;RELOCATE THE PROGRAM FIRST TO THE NEXT HIGHER B
2585  031354  012737  000001  014532          MOV     #1,     HICORE  ;TURN MM ON ON POWER-UP
2586  031362  012737  000001  014534          MOV     #1,     RELOUP  ;SIGNAL SLAVES TO RELOCATE
2587  031370  063737  014550  172340          ADD     HIBOX,  KIPAR0  ;GET READY TO GO TO HIGH CORE
2588  031376  063737  014550  172342          ADD     HIBOX,  KIPAR1
2589  031404  063737  014550  172344          ADD     HIBOX,  KIPAR2
2590  031412  052737  000001  177572          BIS     #1,     MMR0    ;WE ARE NOW IN HIGH CORE
2591  031420                          10$:
2592  031420  005037  014530          11$:    CLR     PATCHK          ;SET TO WRITE PATTERN
2593  031424  004737  032434                  JSR     PC,     PATTRN  ;WRITE THE PATTERN
2594  031430  004737  032322                  JSR     PC,     BUFCLR  ;CLEAR THE KEYBOARD BUFFER
2595  031434  104401  042317                  TYPE    ,TM106          ;TELL THE OPERATOR TO POWER FAIL THE MEMORY BOX
2596  031440  006202                          ASR     R2
2597  031442  010246                          MOV     R2,     -(SP)
2598  031444  104405                          TYPDS
2599  031446  006302                          ASL     R2
2600  031450  104401  044053                  TYPE    ,TM110          ;SPECIFY THE CONDITIONS
2601  031454  012703  031616                  MOV     #100$,  R3      ;SET UP THE POWER FAIL RETURN
2602  031460  005737  014526                  TST     BOOT            ;IS THIS THE DC TEST?
2603  031464  001414                          BEQ     12$             ;BRANCH IF NO
2604  031466  012737  020000  000000  13$:     MOV     #20000, @#0     ;MAKE BOOTING SLAVE SP 20000
```

```
2605  031474  005237  000110          INC     @#110                   ;HANG THE SLAVES BOOT
2606  031500  105777  162504          TSTB    @$TKS                   ;IS CHARACTER IN BUFFER?
2607  031504  100370                  BPL     13$                     ;LOOP
2608  031506  012777  032762  165312  MOV     #ENTR,  @ISTVEC         ;SET UP TO INTERRUPT SLAVES
2609  031514  000413                  BR      14$                     ;CONTINUE
2610  031516                  12$:
2611  031516  105777  162466          TSTB    @$TKS                   ;IS CHARACTER IN THE BUFFER?
2612  031522  100375                  BPL     12$                     ;LOOP
2613  031524  005737  014522  17$:    TST     PWRFL                   ;SHOULD THE MASTER HAVE POWER FAILED?
2614  031530  001402                  BEQ     18$                     ;BRANCH IF NO
2615  031532  104024                  ERROR   24                      ;FAILURE TO POWER FAIL
2616  031534  000441                  BR      102$                    ;CONTINUE
2617  031536  104401  047036  18$:    TYPE    ,OK
2618  031542  000436                  BR      102$
2619  031544  017704  162410  14$:    MOV     @SWR,   R4              ;GET THE SWITCH VALUES
2620  031550  042704  177760          BIC     #177760,        R4      ;SAVE ONLY CPU BITS
2621  031554  012777  000000  165240  MOV     #PGTE,  @ACR            ;ACCESS PGTE REG
2622  031562  010477  165236          MOV     R4,     @ADR            ;SET INTERRUPT BITS
2623  031566  032737  000001  016766  BIT     #BIT0,  CPUACT          ;EVEN OR ODD?
2624  031574  001404                  BEQ     15$                     ;BRANCH IF EVEN
2625  031576  012777  000001  165220  MOV     #1,     @ADR            ;INTERRUPTING AN EVEN NUMBER OF SLAVES
2626  031604  000422                  BR      103$                    ;
2627  031606  012777  000003  165210  15$:    MOV     #3,     @ADR            ;INTERRUPTING AN ODD # OF SLAVES
2628  031614  000416                  BR      103$
2629
2630  031616  005737  014522  100$:   TST     PWRFL                   ;DID WE EXPECT POWER FAIL?
2631  031622  001002                  BNE     101$                    ;BRANCH IF YES
2632  031624  104001                  ERROR   1                       ;MASTER ERROREOUSLY POWER FAILED
2633  031626  000404                  BR      102$                    ;CONTINUE
2634  031630  005060  017004  101$:   CLR     NOPRMP(R0)              ;ALLOW CPU IDENTIFICATION
2635  031634  104401  041635          TYPE    ,TM103                  ;CPU POWER FAIL MSG
2636  031640  012737  000001  014530  102$:   MOV     #1,     PATCHK          ;SET UP FOR PATTERN CHECK
2637  031646  004737  032434          JSR     PC,     PATTRN          ;CHECK THE PATTERN
2638  031652  032737  000001  177572  103$:   BIT     #1,     MMR0            ;ARE WE IN HIGH CORE?
2639  031660  001411                  BEQ     104$                    ;BRANCH IF NO
2640  031662  004737  032722          JSR     PC,     RELOLO          ;ELSE RELOCATE
2641  031666  012737  000001  014536  MOV     #1,     RELODN          ;SIGNAL THE SLAVES
2642  031674  005037  177572          CLR     MMR0                    ;WE ARE NOW BACK DOWN IN LOW CORE
2643  031700  005037  014532          CLR     HICORE                  ;MAKE SURE MM ON POWER-UP DISABLED
2644  031704  005005                  CLR     R5                      ;MAKE TS24A DELAY
2645  031706  077501                  SOB     R5,     .
2646  031710  162702  000002          SUB     #2,     R2              ;POINT TO NEXT BOX
2647  031714  000137  031232          JMP     2$                      ;CONTINUE
2648
2649
2650
```

```
                                ;;*****************************************************************
                                ;*TEST 25       CHECK SYSTEM RECOVERY ON AC POWER FAIL
                                ;;*****************************************************************
031720                          TST25:
031720  012777  000025  162242          MOV     #25,@DISPLAY    ;SET TEST NUMBER
031726  112761  000025  014002          MOVB    #25,    $TSTNM(R1)      ;SET THE TEST NUMBER
031734  032770  000400  014160          BIT     #SW08,  @SWR(R0)        ;SKIP THIS TEST?
031742  001002                        BNE     1$                      ;BRANCH IF NO
031744  000137  035060                JMP     $EOP            ;ELSE GO TO EOP
031750                          1$:
031750  020027  000000                CMP     R0,#0           ;SWITCH THIS PROCESSOR?
031754  001002                        BNE     64$             ;BRANCH IF NO
031756  000137  031766                JMP     TS25A           ;YES, SWITCH TO A.
031762  000137  032020          64$:  JMP     TS25B           ;NO, SWITCH TO B.
031766                          TS25A:
031766  005700                        TST     R0              ;IS THIS THE MASTER?
031770  001007                        BNE     5$              ;BRANCH IF NO
031772  104401  041476                TYPE    ,TM77           ;'TEST'
031776  005046                        CLR     -(SP)
032000  116116  014002                MOVB    $TSTNM(R1),(SP) ;GET THE TEST NO.
032004  104403                        TYPOS
032006  000002                        .WORD   2                       ;TYPE 2 DIGITS, NO LEADING 0
032010                          5$:
032010  104401  014333                TYPE    ,$CRLF
032014  104401  041700                TYPE    ,TM104          ;'POWER FAIL ENTIRE SYSTEM'
032020  012760  004000  014612  TS25B:MOV     #TI,    PFFT(R0)
032026  005003                        CLR     R3
032030  005037  014714                CLR     SYNC.2
032034  000001                        WAIT                            ;WAIT FOR POWER TO FAIL
032036  106237  016770          3$:   ASRB    SYNC.3                  ;CONTROL THE CPUS
032042  103375                        BCC     3$
032044  005237  014714                INC     SYNC.2                  ;COUNT THE CPUS
032050  012737  000001  016770        MOV     #1,     SYNC.3          ;ALLOW ANOTHER IN
032056  023737  016766  014714  1$:   CMP     CPUACT, SYNC.2
032064  001374                        BNE     1$
032066  104401  014220                TYPE    ,$NULL
032072  000137  035060                JMP     $EOP                    ;GO TO END OF PASS
                                
                                        .SBTTL  MEMORY BOX TEST ROUTINES
032076  005002                  MEMSIZ: CLR     R2                      ;GET SET TO FILL THE START AND STOP TABLES
032100  012704  172100                MOV     #172100,        R4      ;POINT TO THE FIRST CSR
032104  005037  014646                CLR     BOXNUM                  ;START WITH 0 BOXES
032110                          1$:
032110  012760  032232  014702        MOV     #100$,  ERRTAB(R0)      ;SET UP FOR NO BOX
032116  052714  000010                BIS     #10,    (R4)            ;SET UP TO GET BOX CAPACITY
032122  012462  014462                MOV     (R4)+,  STOP(R2)
032126  000362  014462                SWAB    STOP(R2)
032132  042762  177600  014462        BIC     #177600,        STOP(R2);NOW WE HAVE IT
032140  012462  014502                MOV     (R4)+,  START(R2)       ;NOW GET THE STARTING ADR.
032144  042762  177000  014502        BIC     #177000,        START(R2)
032152  016205  014502                MOV     START(R2),      R5      ;MAKE IT LOOK LIKE A PAR
032156  072527  000012                ASH     #10.,   R5
032162  010562  014502                MOV     R5,     START(R2)
032166  016205  014462                MOV     STOP(R2),       R5      ;DO THE SAME FOR STOP
032172  072527  000012                ASH     #10.,   R5
032176  010562  014462                MOV     R5,     STOP(R2)
```

```
2707  032202  066262  014502  014462          ADD     START(R2),      STOP(R2);STOP=START+CAPACITY
2708  032210  005237  014646                  INC     BOXNUM                  ;INCREMENT BOX COUNT
2709  032214  005762  014502                  TST     START(R2)               ;IS THIS THE BASE BOX?
2710  032220  001413                          BEQ     101$                    ;BRANCH IF YES
2711  032222  016237  014502  014550          MOV     START(R2),      HIBOX   ;GET A BOX TO RELOCATE TO
2712  032230  000407                          BR      101$
2713  032232  012762  177777  014502  100$:   MOV     #-1,    START(R2)       ;INDICATE NON-EXISTENT BOX
2714  032240  062706  000004                  ADD     #4,     SP              ;RESTORE THE STACK
2715  032244  062704  000004                  ADD     #4,     R4              ;POINT TO NEXT CSR PAIR
2716  032250  062702  000002          101$:   ADD     #2,     R2              ;POINT TO NEXT TABLE LOCATIONS
2717  032254  022702  000020                  CMP     #20,    R2              ;HAVE WE LOOKED FOR EIGHT BOXES?
2718  032260  003313                          BGT     1$                      ;BRANCH IF NO
2719  032262  012760  040362  014702          MOV     #CPUER, ERRTAB(R0)      ;RESET TRAP TO 4 POINTER
2720  032270  022737  000001  014646          CMP     #1,     BOXNUM          ;WAS THERE MORE THAN A SINGLE BOX?
2721  032276  002410                          BLT     102$                    ;BRANCH IF YES
2722  032300  005737  014522                  TST     PWRFL                   ;IS THIS TEST 23?
2723  032304  001005                          BNE     102$                    ;BRANCH IF YES
2724  032306  104401  043771                  TYPE    ,TM109                  ;PRINT ONLY ONE BOX MSG
2725  032312  012737  000001  014540          MOV     #1,     EXIT            ;SIGNAL TO EXIT
2726
2727  032320  000207                  102$:   RTS     PC                      ;RETURN
2728
2729  032322  105777  161662          BUFCLR: TSTB    @$TKS                   ;IS THE BUFFER EMPTY?
2730  032326  100003                          BPL     1$                      ;BRANCH IF YES
2731  032330  117705  161656                  MOVB    @$TKB,  R5              ;FLUSH IT
2732  032334  000772                          BR      BUFCLR                  ;LOOP
2733  032336  000207                  1$:     RTS     PC                      ;RETURN
2734
2735  032340  005037  172340          SETMM:  CLR     KIPAR0                  ;SET UP PARS 0,1,2,7
2736  032344  012737  077406  172300          MOV     #77406, KIPDR0
2737  032352  012737  000200  172342          MOV     #200,   KIPAR1
2738  032360  012737  077406  172302          MOV     #77406, KIPDR1
2739  032366  012737  000400  172344          MOV     #400,   KIPAR2
2740  032374  012737  077406  172304          MOV     #77406, KIPDR2
2741  032402  012737  177600  172356          MOV     #177600,        KIPAR7
2742  032410  012737  077406  172316          MOV     #77406, KIPDR7
2743  032416  012737  000020  172516          MOV     #20,    MMR3
2744  032424  012737  033270  000250          MOV     #MMERR, MMVEC
2745  032432  000207                          RTS     PC
2746
2747  032434  005737  014526          PATTRN: TST     BOOT                    ;ARE WE DOING THE DC TEST?
2748  032440  001055                          BNE     7$                      ;BRANCH IF YES
2749  032442  005762  014502                  TST     START(R2)               ;ARE WE DOING THE BASE BOX?
2750  032446  001452                          BEQ     7$                      ;BRANCH IF YES
2751  032450  004737  032340                  JSR     PC,     SETMM           ;SET UP PARS 0,1,2,7
2752  032454  016237  014502  172346          MOV     START(R2),      KIPAR3  ;USE PAR3 TO WRITE PATTRN
2753  032462  012737  077406  172306          MOV     #77406, KIPDR3
2754  032470  012760  040362  014702          MOV     #CPUER, ERRTAB(R0)      ;SET UP FOR UNEXPECTED TRAPS
2755  032476  052737  000001  177572          BIS     #1,     MMR0            ;TURN ON MM
2756  032504  012705  060000          1$:     MOV     #60000, R5              ;POINT TO PAR3 SPACE
2757  032510  026237  014462  172346          CMP     STOP(R2),       KIPAR3  ;IS THIS THE END OF THE BOX?
2758  032516  003424                          BLE     6$                      ;BRANCH IF YES
2759  032520  005737  014530          2$:     TST     PATCHK                  ;ARE WE WRITING A PATTERN?
2760  032524  001002                          BNE     3$                      ;BRANCH IF ONLY READING IT
2761  032526  012715  152525                  MOV     #152525,        (R5)    ;WRITE THE PATTERN
2762  032532  022725  152525          3$:     CMP     #152525,        (R5)+   ;IS THE PATTERN CORRECT?
```

```
2763  032536  001405                    BEQ     4$                      ;BRANCH IF YES
2764  032540  013760  172346  014250    MOV     KIPAR3, $REG1(R0)       ;SAVE THE BAD ADDRESS (PAR)
2765  032546  104013                    ERROR   13                      ;MEMORY IS CORRUPTED
2766  032550  000407                    BR      6$                      ;REPORT ONLY ONE ERROR
2767  032552  022705  100000          4$: CMP     #100000,        R5      ;ARE WE STILL IN PAR3 SPACE?
2768  032556  003360                    BGT     2$                      ;BRANCH IF YES
2769  032560  062737  000200  172346    ADD     #200,   KIPAR3          ;ELSE RESET THE PAR
2770  032566  000746                    BR      1$                      ;AND CONTINUE
2771  032570                          6$:
2772  032570  005037  177572            CLR     MMR0                    ;END MM
2773  032574  000207                  7$: RTS     PC                      ;RETURN
2774
2775  032576  004737  032340          RELOHI: JSR     PC,     SETMM           ;SET UP PARS 0,1,2,7
2776  032602  012760  040362  014702        MOV     #CPUER, ERRTAB(R0)      ;SET UP FOR TRAP TO 4
2777  032610  013737  014550  172346        MOV     HIBOX,  KIPAR3          ;PARS 3,4,5 WILL TAKE US TO HIGH CORE
2778  032616  012737  077406  172306        MOV     #77406, KIPDR3
2779  032624  013737  014550  172350        MOV     HIBOX,  KIPAR4
2780  032632  062737  000200  172350        ADD     #200,   KIPAR4
2781  032640  012737  077406  172310        MOV     #77406, KIPDR4
2782  032646  013737  014550  172352        MOV     HIBOX,  KIPAR5
2783  032654  062737  000400  172352        ADD     #400,   KIPAR5
2784  032662  012737  077406  172312        MOV     #77406, KIPDR5
2785  032670  012704  060000                MOV     #60000, R4              ;POINT TO PAR3
2786  032674  005005                        CLR     R5                      ;POINT TO PAR0
2787  032676  052737  000001  177572        BIS     #1,     MMR0            ;TURN ON MM
2788  032704  012524                  1$: MOV     (R5)+,  (R4)+           ;RELOCATE THE PROGRAM
2789  032706  022705  060000                CMP     #60000, R5              ;ARE WE FINISHED?
2790  032712  003374                        BGT     1$                      ;BRANCH IF NO
2791  032714  005037  177572                CLR     MMR0                    ;TURN OFF MM
2792  032720  000207                        RTS     PC
2793
2794  032722  005037  172346          RELOLO: CLR     KIPAR3                  ;USE PARS 3,4,5 TO RESTORE CODE TO LOW CORE
2795  032726  012737  000200  172350        MOV     #200,   KIPAR4
2796  032734  012737  000400  172352        MOV     #400,   KIPAR5
2797  032742  012704  060000                MOV     #60000, R4              ;POINT TO PAR 3
2798  032746  005005                        CLR     R5                      ;POINT TO PAR 0
2799  032750  012524                  1$: MOV     (R5)+,  (R4)+           ;RESTORE THE PROGRAM
2800  032752  022705  060000                CMP     #60000, R5              ;ARE WE FINISHED?
2801  032756  003374                        BGT     1$                      ;BRANCH IF NO
2802  032760  000207                        RTS     PC                      ;RETURN-STILL EXECUTING IN HIGH CORE
2803
2804  032762  052737  001000  177746  ENTR:   BIS     #BIT9,  CONTRL  ;TURN OFF CACHE
2805  032770  017705  164026                MOV     @ACR,   R5              ;GET CPU ID
2806  032774  072527  177770                ASH     #-10,   R5
2807  033000  005004                        CLR     R4                      ;SET UP R0
2808  033002  026405  014226          65$:  CMP     $CPUID(R4),     R5
2809  033006  001404                        BEQ     64$
2810  033010  005724                        TST     (R4)+
2811  033012  020427  000010                CMP     R4,     #10
2812  033016  002771                        BLT     65$
2813  033020  010400                  64$:  MOV     R4,     R0
2814  033022  010001                        MOV     R0,     R1              ;SET UP R1
2815  033024  006201                        ASR     R1
2816  033026  052777  100000  163766        BIS     #BIT15, @ACR            ;INITIALIZE THE IIST
2817  033034  016006  014200                MOV     $$STP(R0),      SP      ;SET UP THE STACK
2818  033040  012760  034412  014662        MOV     #$PWRDN,        PWRTAB(R0);SET UP FOR POWER DOWN
```

```
 2819  033046  020027  000000                CMP     R0,     #0              ;SHOULD WE BE HERE?
 2820  033052  001406                        BEQ     1$                      ;BRANCH IF NO
 2821  033054  005060  017004                CLR     NOPRMP(R0)
 2822  033060  104401  044746                TYPE    .TM111
 2823  033064  000137  030762                JMP     1S24B                   ;RETURN SLAVES
 2824  033070  104025                1$:     ERROR   25                      ;UNEXPECTED INTERRUPT
 2825  033072  000000                        HALT                            ;HALT THE MASTER
```

```
2826                                        .SBTTL  PARITY ERROR HANDLER
2827
2828  033074  005327                PARERR: DEC     (PC)+           ;FIRST TIME IN?
2829  033076  000001                PARFLG: .WORD   1
2830  033100  002001                        BGE     1$                      ;BRANCH IS YES
2831  033102  000000                        HALT                            ;ELSE HALT
2832  033104  013760  177766  014250  1$:   MOV     CPUERR, $REG1(R0)
2833  033112  013760  177744  014260        MOV     MEMERR, $REG2(R0)
2834  033120  104007                        ERROR   7                       ;UNEXPECTED TRAP TO 114
2835  033122  000000                        HALT
2836  033124  013737  177744  177744        MOV     @#MEMERR,@#MEMERR ;CLEAR ERROR INDICATORS
2837  033132  012737  000001  033076        MOV     #1,PARFLG       ;INITIALIZE PARITY ERROR FLAG
2838  033140  000002                        RTI                     ;CONTINUE TESTING
2839                                        .SBTTL  SETUP MEMORY MANAGMENT REGISTERS
2840
2841  033142  012737  000000  172340  MAP:  MOV     #0,@#KIPAR0     ;SETUP PAR0 FOR 1ST 4K
2842  033150  012737  077406  172300        MOV     #77406,@#KIPDR0 ;4K, R/W, EXPAND UP
2843  033156  012737  000200  172342        MOV     #200,@#KIPAR1   ;SETUP PAR0 FOR 2ND 4K
2844  033164  012737  077406  172302        MOV     #77406,@#KIPDR1 ;4K,R/W,EXPAND UP
2845  033172  012737  000400  172344        MOV     #400,@#KIPAR2   ;SETUP PAR2 FOR NEXT 4K
2846  033200  012737  077406  172304        MOV     #77406,@#KIPDR2 ;4K,R/W,EXPAND UP
2847  033206  012737  000000  172352        MOV     #0,@#KIPAR5     ;SET UP PAR5 FOR 1ST 4K
2848  033214  012737  077406  172312        MOV     #77406,@#KIPDR5 ;4K, R/W, ED=UP
2849  033222  012737  000200  172354        MOV     #200,@#KIPAR6   ;SET UP PAR6 FOR 2ND 4K
2850  033230  012737  000000  172314        MOV     #0,@#KIPDR6     ;ABORT ALL REFERENCES
2851  033236  012737  177600  172356        MOV     #177600,@#KIPAR7  ;SET UP PAR7 FOR I/O PAGE
2852  033244  012737  077406  172316        MOV     #77406,@#KIPDR7 ;4K, R/W, ED=UP
2853  033252  012737  000020  172516        MOV     #BIT04,@#MMR3   ;SET UP FOR 22-BIT MAPPING
2854  033260  012737  033270  000250        MOV     #MMERR,@#MMVEC  ;SET UP MEMORY MANAGEMENT VECTOR
2855  033266  000207                        RTS     PC              ;RETURN FROM CALL
2856  033270  000000                MMERR:  HALT                    ;MEMORY MANAGEMENT ERROR
2857  033272  000776                        BR      MMERR
2858
2859                                        .SBTTL  MASSBUS TRANSFER ROUTINES
2860  033274                        MBUSR:
2861  033300  032737  000200  176700  4$:   BIT     #BIT7,  @#RPCS1         ;WAIT FOR CONTROLLER READY
2862  033302  001774                        BEQ     4$
2863  033304  156037  014632  176710        BISB    MBD.W(R0),      @#RPCS2 ;GET THE DRIVE #
2864  033312  012737  174000  176702        MOV     #-4000,.#RPWC           ;;SET WORD COUNT
2865  033320  010437  176704                MOV     R4,@#RPBA               ;;SET MEMORY ADDRESS
2866  033324  005037  176706                CLR     @#RPDA                  ;;READ SECTOR 0
2867  033330  032737  000200  176712  3$.   BIT     #BIT7,  @#RPDS          ;WAIT FOR DRIVE READY
2868  033336  001774                        BEQ     3$
2869  033340  052737  000001  176700        BIS     #BIT0,@#RPCS1           ;;
2870  033346  106237  033270          1$:   ASRB    MMERR                   ;;DO ASRB DURING TRANSFER
2871  033352  106237  033270                ASRB    MMERR
2872  033356  106237  033270                ASRB    MMERR
2873  033362  106237  033270                ASRB    MMERR
2874  033366  106237  033270                ASRB    MMERR
2875  033372  106237  033270                ASRB    MMERR
2876  033376  106237  033270                ASRB    MMERR
2877  033402  106237  033270                ASRB    MMERR
2878  033406  106237  033270                ASRB    MMERR
2879  033412  106237  033270                ASRB    MMERR
2880  033416  032737  000200  176712        BIT     #BIT7,@#RPDS            ;;DEVICE READY?
2881  033424  001750                        BEQ     1$                      ;;BRANCH IF NO.
```

```
2882  033426  005737  176700          TST     @#RPCS1                 ;;ANY ERRORS?
2883  033432  100001                  BPL     2$                      ;;NO
2884  033434  000000                  HALT                            ;;YES
2885  033436  000207          2$:     RTS     PC                      ;;RETURN
2886                                  .SBTTL  LINE CLOCK ROUTINE
2887  033440  012737  033460  000100  SETCLK: MOV     #5$,@#100               ;;SET THE INTERRUPT VECTOR FOR CLK
2888  033446  005204                  INC     R4                      ;;ADD 1 TO THE ARGUMENT PASSED
2889  033450  012737  000100  177546          MOV     #BIT6,@#LKS             ;;START THE CLOCK
2890  033456  000207                  RTS     PC                      ;;RETURN
2891  033460  052737  000340  177776  5$:     BIS     #340,@#PS               ;;HIGH PRIORITY
2892  033466  042737  000200  177546          BIC     #BIT7,@#LKS             ;;CLEAR THE MONITOR BIT
2893  033474  005304                  DEC     R4                      ;;ONE TICK
2894  033476  005704                  TST     R4                      ;;COUNT TO ZERO?
2895  033500  001010                  BNE     6$                      ;;NO DON'T STOP THE CLOCK
2896  033502  005037  177546          CLR     @#LKS                   ;;TURN IT OFF
2897  033506  000240                  NOP
2898  033510  000240                  NOP
2899  033512  000240                  NOP
2900  033514  000240                  NOP
2901  033516  062716  000004          ADD     #4,(SP)                 ;;SKIP RETURN
2902  033522  162716  000002  6$:     SUB     #2,(SP)                 ;;IF COUNT ISN'T EXPIRED...
2903  033526  000002          7$:     RTI                             ;;RETURN TO THE WAIT
```

```
                                .SBTTL  POWER FAIL ROUTINE (SECTION 1)

033530  012737  034172  014662  POWDWN: MOV     #ILLUP,PWRTAB   ;IF TOO FAST WITH POWER UP
033536  105737  016740                  TSTB    UBEF            ;;USE UNIBUS EXERCISER?
033542  001403                          BEQ     64$
033544  042737  000020  170016          BIC     #BIT4,@#UBCR2   ;;CLEAR POWER FAIL ENABLE
033552                          64$:
033552  022706  000440                  CMP     #440,SP         ;YELLOW OR RED?
033556  100402                          BMI     1$              ;NO
033560  016006  014200                  MOV     $$STP(R0),SP    ;SET EMERGENCY STACK
033564  010246                  1$:     MOV     R2,-(SP)                ;SAVE R2
033566  012702  177572                  MOV     #MMR0,R2                ;SAVE PSW THRU MMR0
033572  012246                  10$:    MOV     (R2)+,-(SP)
033574  022702  177676                  CMP     #UDPAR7,R2
033600  103374                          BHIS    10$
033602  013746  177776                  MOV     @#PSW,-(SP)     ;SAVE PSW
033606  013746  177746                  MOV     @#CONTRL,-(SP)  ;SAVE CACHE CONTROL
033612  013746  172516                  MOV     @#MMR3,-(SP)            ;SAVE MMR3
033616  012702  172200                  MOV     #SIPDR0,R2              ;SAVE SIPDR0 THRU KDPAR7
033622  012246                  20$:    MOV     (R2)+,-(SP)
033624  022702  172376                  CMP     #KDPAR7,R2
033630  103374                          BHIS    20$
033632  012702  170200                  MOV     #MAPL00,R2              ;SAVE THE UNIBUS MAP
033636  012246                  30$:    MOV     (R2)+,-(SP)
033640  022702  170376                  CMP     #MAPH37,R2
033644  103374                          BHIS    30$
033646  010046                          MOV     R0,-(SP)        ;SAVE THE GENERAL REGISTERS
033650  010146                          MOV     R1,-(SP)
033652  010346                          MOV     R3,-(SP)
033654  010446                          MOV     R4,-(SP)
033656  010546                          MOV     R5,-(SP)
033660  010637  014572                  MOV     SP,     SAV6    ;SAVE THE STACK
033664  005060  016754                  CLR     COUNT0(R0)      ;CLEAR LOOP COUNTER
033670  060000                          ADD     R0,     R0      ;       INDEX TO THE RIGHT COUNTER
033672  060000                          ADD     R0,     R0
033674  060000                          ADD     R0,R0
033676  160100                          SUB     R1,R0
033700  160100                          SUB     R1,R0
033702  012737  034012  014662          MOV     #POWUP, PWRTAB  ;ENABLE GOOD POWER-UP
033710  012737  000001  016752          MOV     #1,LOOPS        ;# OF LOOPS FOR 2 MS (WORST CASE CONTENTION)
033716  000160  033722                  JMP     3$(R0)          ;START LOOPING UNTIL DC POWER FAILS
033722  005237  016754          3$:     INC     COUNT0
                                
033726  023737  016752  016754          CMP     LOOPS,COUNT0            ;2MS UP?
033734  001372                          BNE     3$                      ;BRANCH IF NO
033736  000000                          HALT                            ;FINISHED
033740  005237  016756          4$:     INC     COUNT1
033744  023737  016752  016756          CMP     LOOPS,COUNT1
033752  001372                          BNE     4$
033754  000000                          HALT
033756  005237  016760          5$:     INC     COUNT2
033762  023737  016752  016760          CMP     LOOPS,COUNT2
033770  001372                          BNE     5$
033772  000000                          HALT
033774  005237  016762          6$:     INC     COUNT3
034000  023737  016752  016762          CMP     LOOPS,COUNT3
```

```
2960  034006  001372                    BNE     6$
2961  034010  000000                    HALT
2962
2963  034012                    POWUP:
2964  034012  012737  034176  014662    MOV     #ILLDWN,PWRTAB;SET TOO FAST DOWN VECTOR
2965  034020  013706  014572            MOV     SAV6,SP         ;RESET SP
2966  034024  012605                    MOV     (SP)+,R5        ;RESTORE THE REGISTERS
2967  034026  012604                    MOV     (SP)+,R4
2968  034030  012603                    MOV     (SP)+,R3
2969  034032  012601                    MOV     (SP)+,R1
2970  034034  012600                    MOV     (SP)+,R0
2971  034036  012702  170400            MOV     #MAPH37+2,R2    ;RESTORE UNIBUS MAP
2972  034042  012642            10$:    MOV     (SP)+,-(R2)
2973  034044  022702  170200            CMP     #MAPL00,R2
2974  034050  103774                    BLO     10$
2975  034052  012702  172400            MOV     #KDPAR7+2,R2    ;RESTORE K AND S PARS/PDRS
2976  034056  012642            20$.    MOV     (SP)+,-(R2)
2977  034060  022702  172200            CMP     #SIPDR0,R2
2978  034064  103774                    BLO     20$
2979  034066  012637  172516            MOV     (SP)+,@#MMR3    ;RESTORE MMR3
2980  034072  012637  177746            MOV     (SP)+,@#CONTRL  ;RESTORE CACHE CONTRL
2981  034076  012637  177776            MOV     (SP)+,@#PSW     ;RESTORE PSW
2982  034102  012702  177700            MOV     #UDPAR7+2,R2    ;RESTORE PSW THRU MMR0
2983  034106  012642            30$:    MOV     (SP)+,-(R2)
2984  034110  022702  177572            CMP     #MMR0,R2
2985  034114  103774                    BLO     30$
2986  034116  012602                    MOV     (SP)+,R2        ;RESTORE R2
2987  034120  004737  034202            JSR     PC,     TIMIT   ;CHECK THE POWER-DOWN TIME
2988  034124  012737  033530  014662    MOV     #POWDWN,PWRTAB ;RESET THE DOWN VECTOR
2989  034132  105737  016740            TSTB    UBEF            ;UBE BEING USED?
2990  034136  001403                    BEQ     2$              ;BRANCH IF NO
2991  034140  012737  000001  017014    MOV     #1,UBELCK       ;CLEAR THE PF LOCK
2992  034146                    2$:
2993  034146  105737  016737            TSTB    MPF             ;MULTIPROCESSOR MODE?
2994  034152  001403                    BEQ     1$              ;BRANCH IF NO
2995  034154  052737  001000  177746    BIS     #1000,@#CONTRL  ;BYPASS CACHE
2996  034162  052737  000014  177746  1$:BIS    #14,    @#CONTRL        ;TURN OFF CACHE
2997  034170  000113                    JMP     (R3)            ;JUMP INDIRECT TO R3
2998
2999  034172  000000            ILLUP:  HALT                    ;POWER UP BEFORE POWER DOWN COMPLETE
3000  034174  000776                    BR      .-2             ;LOCK UP THE HALT
3001
3002  034176  000000            ILLDWN: HALT                    ;POWERED DOWN BEFORE UP COMPLETE
3003  034200  000776                    BR      .-2             ;LOCK UP THE HALT
3004
3005
3006  034202                    TIMIT:
3007  034202  023760  016752  016754    CMP     LOOPS,COUNT0(R0)        ;DID WE HAVE ENOUGH POWER DOWN TIME?
3008  034210  001402                    BEQ     1$              ;BRANCH IF YES
3009  034212  104401  045004            TYPE    ,$DOWN                  ;NOT ENOUGH TIME
3010
3011  034216  000207            1$:     RTS     PC
```

```
                                        .SBTTL  POWER FAIL ROUTINE (SECTION 2)
034220                          $POWER:
034220  005737  014532                  TST     HICORE                  ;DID WE POWER DOWN IN HIGH CORE?
034224  001451                          BEQ     1$                      ;BRANCH IF NO
034226  005037  172340                  CLR     KIPAR0                  ;SET UP PARS 0,1,2,7
034232  012737  077406  172300          MOV     #77406, KIPDR0
034240  012737  000200  172342          MOV     #200,   KIPAR1
034246  012737  077406  172302          MOV     #77406, KIPDR1
034254  012737  000400  172344          MOV     #400,   KIPAR2
034262  012737  077406  172304          MOV     #77406, KIPDR2
034270  012737  177600  172356          MOV     #177600,        KIPAR7
034276  012737  077406  172316          MOV     #77406, KIPDR7
034304  012737  000020  172516          MOV     #20,    MMR3
034312  012737  033270  000250          MOV     #MMERR, MMVEC
034320  063737  014550  172340          ADD     HIBOX,  KIPAR0
034326  063737  014550  172342          ADD     HIBOX,  KIPAR1
034334  063737  014550  172344          ADD     HIBOX,  KIPAR2
034342  052737  000001  177572          BIS     #1,     MMR0            ;POWER-UP IN HIGH CORE
034350                          1$:
034350  017705  162446                  MOV     @ACR,   R5              ;COPY ACR
034354  072527  177770                  ASH     #-10,   R5              ;GET THE ID
034360  005004                          CLR     R4                      ;SET UP R0
034362  026405  014226          65$:    CMP     $CPUID(R4),     R5
034366  001404                          BEQ     64$
034370  005724                          TST     (R4)+
034372  020427  000010                  CMP     R4,     #10
034376  002771                          BLT     65$
034400  010400                  64$:    MOV     R4,     R0
034402  010001                          MOV     R0,     R1
034404  006201                          ASR     R1
034406  000170  014662                  JMP     @PWRTAB(R0)             ;JUMP TO THE POWER ROUTINE
034412  012760  035050  014662  $PWRDN: MOV     #$ILLUP,PWRTAB(R0)      ;SET UVECT FOR ILLEGAL UP
034420  032760  000040  014612          BIT     #SSD,PFFT(R0)           ;SEND A SIGNAL?
034426  001403                          BEQ     10$                     ;NO
034430  012737  177777  014642          MOV     #-1,    SIGNAL
034436                          10$:
034436  026006  017044                  CMP     YELLIM(R0),SP           ;YELLOW OR RED?
034442  100402                          BMI     1$                      ;NO
034444  016006  014200                  MOV     $$STP(R0),SP            ;SET EMERGENCY STACK
034450  010246                  1$:     MOV     R2,-(SP)                ;SAVE R2
034452  012702  177572                  MOV     #MMR0,R2                ;SAVE PSW THRU MMR0
034456  012246                  100$:   MOV     (R2)+,-(SP)
034460  022702  177676                  CMP     #UDPAR7,R2
034464  103374                          BHIS    100$
034466  013746  177776                  MOV     @#PSW,-(SP)     ;SAVE PSW
034472  013746  177746                  MOV     @#CONTRL,-(SP)  ;SAVE CACHE CONTRL
034476  013746  172516                  MOV     @#MMR3,-(SP)            ;SAVE MMR3
034502  012702  172200                  MOV     #SIPDR0,R2              ;SAVE SIPDR0 THRU KDPAR7
034506  012246                  20$:    MOV     (R2)+,-(SP)
034510  022702  172376                  CMP     #KDPAR7,R2
034514  103374                          BHIS    20$
034516  012702  170200                  MOV     #MAPL00,R2              ;SAVE THE UNIBUS MAP
034522  012246                  30$:    MOV     (R2)+,-(SP)
```

```
3068  034524  022702  170376            CMP     #MAPH37,R2
3069  034530  103374                    BHIS    30$
3070  034532  010046                    MOV     R0,-(SP)        ;SAVE THE GENERAL REGISTERS
3071  034534  010146                    MOV     R1,-(SP)
3072  034536  010346                    MOV     R3,-(SP)
3073  034540  010446                    MOV     R4,-(SP)
3074  034542  010546                    MOV     R5,-(SP)
3075  034544  010660  014572      40$:  MOV     SP,     SAV6(R0)        ;SAVE THE STACK
3076  034550  005060  016754            CLR     COUNT0(R0)              ;CLEAR THE LOOP COUNTER
3077  034554  012760  034676  014662    MOV     #$PWRUP,        PWRTAB(R0);GET SET FOR POWER-UP
3078  034562  060000                    ADD     R0,     R0      ;       INDEX TO THE RIGHT COUNTER
3079  034564  060000                    ADD     R0,     R0
3080  034566  060000                    ADD     R0,R0
3081  034570  160100                    SUB     R1,R0
3082  034572  160100                    SUB     R1,R0
3083  034574  012737  000001  016752    MOV     #1,LOOPS                ;# OF LOOPS FOR 2 MS (WORST CASE CONTENTION)
3084  034602  000160  034606            JMP     3$(R0)          ;START LOOPING UNTIL DC POWER FAILS
3085  034606  005237  016754      3$:   INC     COUNT0
3086
3087  034612  023737  016752  016754    CMP     LOOPS,COUNT0            ;2MS UP?
3088  034620  001372                    BNE     3$                      ;BRANCH IF NO
3089  034622  000000                    HALT                            ;FINISHED
3090  034624  005237  016756      4$:   INC     COUNT1
3091  034630  023737  016752  016756    CMP     LOOPS,COUNT1
3092  034636  001372                    BNE     4$
3093  034640  000000                    HALT
3094  034642  005237  016760      5$:   INC     COUNT2
3095  034646  023737  016752  016760    CMP     LOOPS,COUNT2
3096  034654  001372                    BNE     5$
3097  034656  000000                    HALT
3098  034660  005237  016762      6$:   INC     COUNT3
3099  034664  023737  016752  016762    CMP     LOOPS,COUNT3
3100  034672  001372                    BNE     6$
3101  034674  000000                    HALT
3102
3103  034676  012760  035054  014662  $PWRUP: MOV     #ILLDN,PWRTAB(R0)       ;SET VECTOR FOR FAST DOWN
3104  034704  016006  014572            MOV     SAV6(R0),SP             ;RESTORE STACK
3105  034710  012605                    MOV     (SP)+,R5        ;SAVE THE GENERAL REGISTERS
3106  034712  012604                    MOV     (SP)+,R4
3107  034714  012603                    MOV     (SP)+,R3
3108  034716  012601                    MOV     (SP)+,R1
3109  034720  012600                    MOV     (SP)+,R0
3110  034722  012702  170400            MOV     #MAPH37+2,R2    ;RESTORE UNIBUS MAP
3111  034726  012642            10$:   MOV     (SP)+,-(R2)
3112  034730  022702  170200            CMP     #MAPL00,R2
3113  034734  103774                    BLO     10$
3114  034736  012702  172400            MOV     #KDPAR7+2,R2    ;RESTORE K AND S PARS/PDRS
3115  034742  012642            20$:   MOV     (SP)+,-(R2)
3116  034744  022702  172200            CMP     #SIPDR0,R2
3117  034750  103774                    BLO     20$
3118  034752  012637  172516            MOV     (SP)+,@#MMR3    ;RESTORE MMR3
3119  034756  012637  177746            MOV     (SP)+,@#CONTRL  ;RESTORE CACHE CONTRL
3120  034762  012637  177776            MOV     (SP)+,@#PSW     ;RESTORE PSW
3121  034766  012702  177700            MOV     #UDPAR7+2,R2    ;RESTORE PSW THRU MMR0
3122  034772  012642            30$:   MOV     (SP)+,-(R2)
3123  034774  022702  177572            CMP     #MMR0,R2
```

```
3124  035000  103774                        BLO     30$
3125  035002  012602                        MOV     (SP)+,R2        ;RESTORE R2
3126  035004  004737  034202                JSR     PC,     TIMIT           ;CHECK THE POWER DOWN
3127  035010  032760  000020  014612        BIT     #SSU,PFFT(R0)           ;SEND SIGNALS?
3128  035016  001403                        BEQ     45$                     ;NO
3129  035020  012737  177777  014642        MOV     #-1,    SIGNAL
3130  035026  005037  014712          45$:  CLR     SYNC.1                  ;THIS MAY UNLOCK THE OTHER CPUS
3131  035032  012760  034412  014662        MOV     #$PWRDN,PWRTAB(R0)      ;SET VECTOR FOR POWER FAIL
3132  035040  005703                        TST     R3                      ;IS R3 ZERO?
3133  035042  001401                        BEQ     50$                     ;YES
3134  035044  010316                        MOV     R3,(SP)                 ;FUDGE RETURN ADDRESS ON STACK
3135  035046  000002          50$:  RTI
3136
3137  035050                  $ILLUP:
3138  035050  000000                        HALT                            ;POWER UP BEFORE POWER DOWN COMPLETE
3139  035052  000777                        BR      .                       ;HANG UP THE PROCESSOR
3140
3141  035054                  ILLDN:
3142  035054  000000                        HALT                            ;POWER DOWN BEFORE UP COMPLETE
3143  035056  000777                        BR      .
```

```
                                        .SBTTL  END OF PASS ROUTINE

                                        ;;*****************************************************************
                                        ;*INCREMENT THE PASS NUMBER ($PASS)
                                        ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
                                        ;*TYPE 'END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT
                                        ;                       CPU #0 > AAAAA
                                        ;                       CPU #1 > BBBBB
                                        ;                       CPU #2 > CCCCC
                                        ;                       CPU #3 > DDDDD
                                        ;       TOTAL SYSTEM-WIDE ERRORS YYYYY''
                                        ;*WHERE XXXXX,AAAAA,BBBBB,CCCCC,DDDDD, AND YYYYY ARE DECIMAL NUMBERS
                                        ;*IF THERES A MONITOR GO TO IT
                                        ;*IF THERE ISN'T JUMP TO RESTAB

035060                                  $EOP:
035060  106237  016770                          ASRB    SYNC.3          ;;CONTROL ENTRY
035064  103375                                  BCC     $EOP
035066  105737  016737                          TSTB    MPF             ;;MP MODE?
035072  001416                                  BEQ     4$              ;;BRANCH IF NO
035074  005700                                  TST     R0              ;;IS THIS THE MASTER?
035076  001414                                  BEQ     4$              ;;YES
035100  005237  014134                          INC     $EOPSG
035104  012737  000001  016770                  MOV     #1,     SYNC.3  ;;ALLOW ANOTHER CPU IN
035112  005737  014134          1$:             TST     $EOPSG          ;;IS THE MASTER FINISHED?
035116  001375                                  BNE     1$              ;;BRANCH IF NO
035120  016006  014200                          MOV     $$STP(R0),SP    ;;RESET THE STACK
035124  000170  036616                          JMP     @RESTAB(R0)
035130                                  4$:
035130  005237  014134                          INC     $EOPSG
035134  012737  000001  016770                  MOV     #1,     SYNC.3  ;;ALLOW ANOTHER CPU IN
035142  023737  014134  016766          5$:     CMP     $EOPSG, CPUACT
035150  001374                                  BNE     5$
035152  005237  014344                          INC     $PASS           ;;INCREMENT THE PASS NUMBER
035156  042737  100000  014344                  BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
035164  005327                                  DEC     (PC)+           ;;LOOP?
035166  000001                  $EOPCT: .WORD   1
035170  003402                                  BLE     1$              ;;YES
035172  000137  035654                          JMP     $DOAGN
035176  012737                  1$:     MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
035200  000001                  $ENDCT: .WORD   1
035202  035166                          $EOPCT
035204  104401  035212                  TYPE    ,65$            ;;TYPE ASCIZ STRING
035210  000407                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <12><15>/END PASS #/
035230                          64$:
035230  013746  014344                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                                ;;TYPE PASS NUMBER
035234  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
035236  104401  035244                  TYPE    ,67$            ;;TYPE ASCIZ STRING
035242  000421                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
035306                          66$:
035306  105737  016737                  TSTB    MPF
035312  001524                          BEQ     UNIEOP
035314  104401  014333                  TYPE    ,$CRLF
```

```
3200  035320  104401  035326          TYPE    ,69$            ;;TYPE ASCIZ STRING
3201  035324  000404                  BR      68$             ;;GET OVER THE ASCIZ
3202                          ;;69$:  .ASCIZ  /CPU#0 /<76>
3203  035336                  68$:
3204  035336  005003                  CLR     R3
3205  035340  004737  035760          JSR     PC,EOPLID
3206  035344  016346  014042          MOV     $ERTTL(R3),-(SP)        ;;SAVE $ERTTL(R3) FOR TYPEOUT
3207  035350  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
3208  035352  104401  014333          TYPE    ,$CRLF
3209  035356  104401  035364          TYPE    ,71$            ;;TYPE ASCIZ STRING
3210  035362  000404                  BR      70$             ;;GET OVER THE ASCIZ
3211                          ;;71$:  .ASCIZ  /CPU#1 /<76>
3212  035374                  70$:
3213  035374  012703  000001          MOV     #1,R3
3214  035400  004737  035760          JSR     PC,EOPLID
3215  035404  016346  014042          MOV     $ERTTL(R3),-(SP)        ;;SAVE $ERTTL(R3) FOR TYPEOUT
3216  035410  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
3217  035412  104401  014333          TYPE    ,$CRLF
3218  035416  104401  035424          TYPE    ,73$            ;;TYPE ASCIZ STRING
3219  035422  000404                  BR      72$             ;;GET OVER THE ASCIZ
3220                          ;;73$:  .ASCIZ  /CPU#2 /<76>
3221  035434                  72$:
3222  035434  012703  000002          MOV     #2,R3
3223  035440  004737  035760          JSR     PC,EOPLID
3224  035444  016346  014042          MOV     $ERTTL(R3),-(SP)        ;;SAVE $ERTTL(R3) FOR TYPEOUT
3225  035450  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
3226  035452  104401  014333          TYPE    ,$CRLF
3227  035456  104401  035464          TYPE    ,75$            ;;TYPE ASCIZ STRING
3228  035462  000404                  BR      74$             ;;GET OVER THE ASCIZ
3229                          ;;75$:  .ASCIZ  /CPU#3 /<76>
3230  035474                  74$:
3231  035474  012703  000003          MOV     #3,R3
3232  035500  004737  035760          JSR     PC,EOPLID
3233  035504  016346  014042          MOV     $ERTTL(R3),-(SP)        ;;SAVE $ERTTL(R3) FOR TYPEOUT
3234  035510  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
3235  035512  104401  014333          TYPE    ,$CRLF
3236  035516  104401  035524          TYPE    ,77$            ;;TYPE ASCIZ STRING
3237  035522  000420                  BR      76$             ;;GET OVER THE ASCIZ
3238                          ;;77$:  .ASCIZ  /TOTAL SYSTEMWIDE ERROR COUNT = /
3239  035564                  76$:
3240  035564                  UNIEOP:
3241  035564  013746  014224          MOV     $ERGBL,-(SP)    ;;SAVE $ERGBL FOR TYPEOUT
3242  035570  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
3243  035572  104401  014333          TYPE    ,$CRLF
3244  035576  005037  014224          CLR     $ERGBL
3245  035602  012703  014042          MOV     #$ERTTL,R3      ;CLEAR THE
3246  035606  005023                  CLR     (R3)+           ;ERROR TOTALS.
3247  035610  005023                  CLR     (R3)+
3248  035612  005023                  CLR     (R3)+
3249  035614  005023                  CLR     (R3)+
3250  035616  005013                  CLR     (R3)
3251  035620  000400                  BR      99$             ;SKIP OVER IN SUBROUTINE
3252  035622  000404          99$:    BR      $GET
3253  035624  013702  000042  $GET42: MOV     @#42,R2         ;;GET MONITOR ADDRESS
3254  035630  001411                  BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
3255  035632  000001                  WAIT
```

```
3256  035634  013702  000042          $GET:   MOV     @#42,R2         ;;INSURE R2 CONTAINS THE MONITORS
3257  035640  001405                          BEQ     $DOAGN          ;;RETURN ADDRESS
3258  035642  000005                          RESET                   ;;CLEAR THE WORLD
3259  035644  004712                  $ENDAD: JSR     PC,(R2)         ;;GO TO MONITOR
3260  035646  000240                          NOP                     ;;SAVE ROOM
3261  035650  000240                          NOP                     ;;FOR
3262  035652  000240                          NOP                     ;;ACT11
3263  035654  013746  014660          $DOAGN: MOV     $PSWR,  -(SP)
3264  035660  013746  014716                  MOV     TYPQUE, -(SP)
3265  035664  013746  014720                  MOV     TYPQUE+2,       -(SP)
3266  035670  013746  016740                  MOV     UBEF,   -(SP)
3267  035674  013746  016766                  MOV     CPUACT, -(SP)
3268  035700  013746  016736                  MOV     FLAGB,  -(SP)
3269  035704  004737  020000                  JSR     PC,     RESTRT
3270  035710  012637  016736                  MOV     (SP)+,  FLAGB
3271  035714  012637  016766                  MOV     (SP)+,  CPUACT
3272  035720  012637  016740                  MOV     (SP)+,  UBEF
3273  035724  012637  014720                  MOV     (SP)+,  TYPQUE+2
3274  035730  012637  014716                  MOV     (SP)+,  TYPQUE
3275  035734  012637  014660                  MOV     (SP)+,  $PSWR
3276  035740  005037  014134                  CLR     $EOPSG          ;;CLEAR THE COUNT AND FREE SLAVES
3277  035744  016006  014200                  MOV     $$STP(R0),SP    ;;RESET THE STACK
3278  035750  000170  036616                  JMP     @RESTAB(R0)             ;RETURN
3279  035754     377     377  000  $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
3280          035760                  .EVEN
3281  035760                  EOPLID:
3282  035760  005002                  CLR     R2              ;RESET FOR COUNT
3283  035762  026203  014226          65$:    CMP     $CPUID(R2),R3   ;SID MATCH?
3284  035766  001404                          BEQ     64$
3285  035770  005722                          TST     (R2)+           ;INCREMENT R2 BY 2
3286  035772  020227  000010                  CMP     R2,#10
3287  035776  002771                          BLT     65$
3288  036000  010203                  64$:    MOV     R2,R3           ;MOV LOGICAL ID TO 2ND OPERAND
3289  036002  000207                          RTS     PC
```

```
                                        .SBTTL  SCOPE HANDLER ROUTINE

                                        ;;**********************************************************************
                                        ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                        ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW14=1         LOOP ON TEST
                                        ;*SW09=1         LOOP ON ERROR
                                        ;*CALL
                                        ;*      SCOPE           ;;SCOPE=IOT

036004                                  $SCOPE:
036004  032770  040000  014160          1$:     BIT     #BIT14,@SWR(R0) ;;LOOP ON PRESENT TEST?
036012  001056                                  BNE     $OVER           ;;YES IF SW14=1
                                        ;#####START OF CODE FOR THE XOR TESTER#####
036014  000416                          $XTSTR: BR      6$              ;;IF RUNNING ON THE 'XOR' TESTER CHANGE
                                                                        ;;THIS INSTRUCTION TO A 'NOP' (NOP-240)
036016  013746  000004                          MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
036022  012737  036042  000004                  MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
036030  005737  177060                          TST     @#177060        ;;TIME OUT ON XOR?
036034  012637  000004                          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
036040  000423                                  BR      $SVLAD          ;;GO TO THE NEXT TEST
036042  022626                          5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
036044  012637  000004                          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
036050  000411                                  BR      7$              ;;LOOP ON THE PRESENT TEST
036052                                  6$:;#####END OF CODE FOR THE XOR TESTER#####
036052  105761  014006                  2$:     TSTB    $ERFLG(R1)      ;;HAS AN ERROR OCCURRED?
036056  001414                                  BEQ     $SVLAD          ;;BR IF NO
036060  032770  001000  014160                  BIT     #BIT09,@SWR(R0) ;;LOOP ON ERROR?
036066  001002                                  BNE     7$              ;;BR IF NO
036070  000160  036104                          JMP     4$(R0)
036074  013760  014032  014022  7$:     MOV     $LPERR,$LPADR(R0)       ;;SET LOOP ADDRESS TO LAST SCOPE
036102  000422                                  BR      $OVER
036104  105061  014006          4$:     CLRB    $ERFLG(R1)      ;;ZERO THE ERROR FLAG
036110  105261  014002          $SVLAD: INCB    $TSTNM(R1)      ;;COUNT TEST NUMBERS
036114  005710                          TST     (R0)            ;;IS THIS THE MASTER
036116  001003                          BNE     1$              ;;NO,
036120  116137  014002  014342          MOVB    $TSTNM(R1),$TESTN       ;;SET TEST NUMBER IN APT MAILBOX
036126                                  1$:
036126  011660  014022                  MOV     (SP),$LPADR(R0) ;;SAVE SCOPE LOOP ADDRESS
036132  011660  014032                  MOV     (SP),$LPERR(R0) ;;SAVE ERROR LOOP ADDRESS
036136  005060  014322                  CLR     $ESCAPE(R0)     ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
036142  112737  000001  014060          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
036150  113771  014002  014170  $OVER:  MOVB    $TSTNM,@DISPLAY(R1) ;;DISPLAY TEST NUMBER
036156  016016  014022                  MOV     $LPADR(R0),(SP) ;;FUDGE RETURN ADDRESS
036162  000002                          RTI                     ;;FIXES PS
```

```
                                .SBTTL  ERROR HANDLER ROUTINE

                                ;;**********************************************************************
                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                ;*MADE BY THE FAILING PROCESSOR
                                ;*AND TYPE OUT THE PROCESSOR ID AND PC OF THE ERROR INSTRUCTION
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW15=1         HALT ON ERROR
                                ;*SW09=1         LOOP ON ERROR
                                ;*CALL
                                ;*      ERROR   N       ;;ERROR=EMT AND N-ERROR ITEM NUMBER

036164                          $ERROR:
036164  105261  014006          7$:     INCB    $ERFLG(R1)      ;;SET THE ERROR FLAG
036170  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
036172  116170  014002  014170          MOVB    $TSTNM(R1),@DISPLAY(R0) ;;DISPLAY TEST NUMBER
036200  005260  014042                  INC     $ERTTL(R0)              ;;INC THE ERROR COUNT
036204  005237  014224                  INC     $ERGBL
036210  011660  014064                  MOV     (SP),$ERRPC(R0) ;;GET ADDRESS OF ERROR INSTRUCTION
036214  162760  000002  014064          SUB     #2,$ERRPC(R0)
036222  117061  014064  014054          MOVB    @$ERRPC(R0),$ITEMB(R1)  ;;STRIP AND SAVE THE ERROR ITEM CODE
036230  010660  014074                  MOV     SP,$ERRSP(R0)   ;;SAVE THE CURRENT STACK POINTER
036234  005770  014160                  TST     @SWR(R0)        ;;HALT ON ERROR?
036240  100025                          BPL     10$             ;;SKIP IF CONTINUE
036242  005700                          TST     R0              ;;IS THIS THE MASTER?
036244  001022                          BNE     3$              ;;NO
036246  104401  036254                  TYPE    ,70$            ;;TYPE ASCIZ STRING
036252  000417                          BR      75$             ;;GET OVER THE ASCIZ
036254  005015  040510  052114  70$:    .ASCIZ  <15><12>/HALT ON MASTER IN $ERROR/<15><12>
036262  047440  020116  040515
036270  052123  051105  044440
036276  020116  042444  051122
036304  051117  005015     000
        036312                          .EVEN
036312                          75$:
036312  000000                  3$:     HALT
036314  000413                  10$:    BR      6$              ;;NO LOOP ON ERROR
036316  032770  001000  014160  4$:     BIT     #BIT9,@SWR(R0)  ;;LOOP ON ERROR SWITCH SET?
036324  001402                          BEQ     5$              ;;BR IF NO
036326  016016  014032                  MOV     $LPERR(R0),(SP) ;;FUDGE RETURN ADDRESS
036332  005760  014322          5$:     TST     $ESCAPE(R0)     ;;CHECK FOR AN ESCAPE ADDRESS
036336  001402                          BEQ     6$              ;;BR IF NONE
036340  016016  014322                  MOV     $ESCAPE(R0),(SP) ;;FUDGE RETURN ADDRESS FOR ESCAPE

036344                          6$:
036344  122737  000001  014356          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
036352  001007                          BNE     11$             ;;NO,SKIP APT ERROR REPORT
036354  116137  014054  036366          MOVB    $ITEMB(R1),21$  ;;SET ITEM NUMBER AS ERROR NUMBER
036362  004737  036644                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
036366     000                  21$:    .BYTE   0
036367     000                          .BYTE   0
036370  000777                  22$:    BR      22$             ;;APT ERROR LOOP
036372                          11$:
036372  022737  035644  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
036400  001001                          BNE     12$             ;;BRANCH IF NO
```

```
3392  036402  000000                          HALT                    ;;YES
3393  036404                          12$:
3394  036404                          $ERRTYP:
3395
3396  036404  106237  016774          64$:    ASRB    ERRLCK
3397  036410  103375                          BCC     64$
3398  036412  010037  017056                  MOV     R0,     TYPLCK          ;ALLOW THIS CPU TO ENTER TYPE ROUTINE
3399  036416  005060  017004                  CLR     NOPRMP(R0)      ;;PRINT MESSAGES WITH PROMPTS
3400  036422  116105  014054                  MOVB    $ITEMB(R1),R5   ;;GET THE ERROR ITEM CODE
3401  036426  072527  000003                  ASH     #3,     R5
3402  036432  162705  000010                  SUB     #10,    R5
3403  036436  062705  017062                  ADD     #$ERRTB,R5      ;;ADD THE ADDR. OF THE ERROR TABLE
3404  036442  011537  036450                  MOV     (R5),1$         ;;SET UP TO...
3405  036446  104401                          TYPE                    ;;TYPE THE ERROR HEADER
3406  036450  000000                  1$:     .WORD   0
3407  036452  104401  014333                  TYPE    ,$CRLF
3408  036456  005725                          TST     (R5)+           ;;INCREMENT R5 BY 2
3409  036460  005715                          TST     (R5)            ;;IS THERE A DATA HEADER?
3410  036462  001406                          BEQ     10$             ;;BRANCH IF NO
3411  036464  011537  036472                  MOV     (R5),2$
3412  036470  104401                          TYPE
3413  036472  000000                  2$:     .WORD   0
3414  036474  104401  014333                  TYPE    ,$CRLF
3415  036500  005725                  10$:    TST     (R5)+           ;;INCREMENT R5
3416  036502  005715                          TST     (R5)            ;;IS THERE DATA TO BE TYPED?
3417  036504  001433                          BEQ     20$
3418  036506  011505                          MOV     (R5),R5         ;;GET THE DATA TABLE ADDRESS
3419  036510  005715                  15$:    TST     (R5)            ;;ARE WE AT THE END OF THE DATA TABLE?
3420  036512  001430                          BEQ     20$             ;;BRANCH IF YES     /
3421  036514  000240                          NOP
3422  036516  011537  014524                  MOV     (R5),   YYY     ;POINT TO THE LOCATION WITH THE NUMBER
3423  036522  022527  014002                  CMP     (R5)+,  #$TSTNM
3424  036526  001405                          BEQ     16$
3425  036530  060037  014524                  ADD     R0,     YYY
3426  036534  017746  155764                  MOV     @YYY,   -(SP)
3427  036540  000405                          BR      17$
3428  036542  005046                  16$:    CLR     -(SP)
3429  036544  060137  014524                  ADD     R1,     YYY
3430  036550  117716  155750                  MOVB    @YYY,   (SP)
3431  036554  104402                  17$:    TYPOC
3432  036556  000240                          NOP
3433  036560  104401  036566                  TYPE    ,66$            ;;TYPE ASCIZ STRING
3434  036564  000402                          BR      65$             ;;GET OVER THE ASCIZ
3435                                  ;;66$:  .ASCIZ  /   /
3436  036572                          65$:
3437  036572  000746                          BR      15$
3438  036574  104401  014333          20$:    TYPE    ,$CRLF
3439  036600  012737  177777  017056          MOV     #-1,    TYPLCK  ;ALLOW ENTRY INTO TYPE ROUTINE
3440  036606  012737  000001  016774          MOV     #1,ERRLCK
3441
3442  036614  000002                          RTI                     ;;RETURN
3443  036616  021420                  RESTAB: TST1                    ;END OF PASS RETURN
3444  036620  021420                          TST1
3445  036622  021420                          TST1
3446  036624  021420                          TST1
3447
```

```
3448
```

```
                                        .SBTTL  APT COMMUNICATIONS ROUTINE

                                        ;;**************************************************************************
036626  112737  000001  037072  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
036634  112737  000001  037070  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
036642  000403                  BR      $ATYC
036644  112737  000001  037072  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
036652                          $ATYC:
036652  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
036654  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
036656  105737  037070          TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
036662  001450                  BEQ     5$              ;;IF NOT:  BR
036664  122737  000001  014356  CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
036672  001031                  BNE     3$              ;;IF NOT:  BR
036674  132737  000100  014357  BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
036702  001425                  BEQ     3$              ;;IF NOT:  BR
036704  017600  000004          MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
036710  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
036716  005737  014336  1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
036722  001375                  BNE     1$              ;;IF NOT:  WAIT
036724  010037  014352          MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
036730  105720          2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
036732  001376                  BNE     2$
036734  163700  014352          SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
036740  006200                  ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
036742  010037  014354          MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
036746  012737  000004  014336  MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
036754  000413                  BR      5$
036756  017637  000004  037002  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
036764  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
036772  013746  177776          MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
036776  004737  037624          JSR     PC,$TYPE        ;;CALL TYPE MACRO
037002  000000          4$:     .WORD   0
037004                  5$:
037004  105737  037072  10$:     TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
037010  001416                  BEQ     12$             ;;IF NOT:  BR
037012  005737  014356          TST     $ENV            ;;RUNNING UNDER APT?
037016  001413                  BEQ     12$             ;;IF NOT:  BR
037020  005737  014336  11$:     TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
037024  001375                  BNE     11$             ;;IF NOT:  WAIT
037026  017637  000004  014340  MOV     @4(SP),$FATAL   ;;GET ERROR #
037034  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
037042  005237  014336          INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
037046  105037  037072  12$:     CLRB    $FFLG           ;;CLEAR FATAL FLAG
037052  105037  037071          CLRB    $LFLG           ;;CLEAR LOG FLAG
037056  105037  037070          CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
037062  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
037064  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
037066  000207                  RTS     PC              ;;RETURN
037070     000  $MFLG:  .BYTE   0       ;;MESSG. FLAG
037071     000  $LFLG:  .BYTE   0       ;;LOG FLAG
037072     000  $FFLG:  .BYTE   0       ;;FATAL FLAG
037074                  .EVEN
000200          APTSIZE=200
000001          APTENV=001
000100          APTSPOOL=100
```

```
3505            000040                  APTCSUP=040
```

```
                                        .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                                        ;;**********************************************************************
                                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                        ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                        ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                        ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                        ;*      .BYTE   M               ;;M=1 OR 0
                                        ;*                                      ;;1=TYPE LEADING ZEROS
                                        ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                        ;*
                                        ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                        ;*$TYPOS OR $TYPOC
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPON                   ;;CALL FOR TYPEOUT
                                        ;*
                                        ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPOC                   ;;CALL FOR TYPEOUT

037074  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
037100  116637  000001  037367          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
037106  112637  037371                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
037112  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
037116  000406                          BR      $TYPON
037120  112737  000001  037367  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
037126  112737  000006  037371          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
037134  112737  000005  037366  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
037142  010346                          MOV     R3,-(SP)        ;;SAVE R3
037144  010446                          MOV     R4,-(SP)        ;;SAVE R4
037146  010546                          MOV     R5,-(SP)        ;;SAVE R5
037150  113737  037371  037372          MOVB    $OMODE+1,DIGITS ;;GET THE NUMBER OF DIGITS TO TYPE
037156  005437  037372                  NEG     DIGITS
037162  062737  000006  037372          ADD     #6,DIGITS               ;;SUBTRACT IT FOR MAX. ALLOWED
037170  113737  037372  037370          MOVB    DIGITS,$OMODE   ;;SAVE IT FOR USE
037176  113737  037367  037372          MOVB    $OFILL,DIGITS   ;;GET THE ZERO FILL SWITCH
037204  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
037210  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
037212  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
037214  000404                          BR      3$              ;;GO DO MSB
037216  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
037220  006105                          ROL     R5
037222  006105                          ROL     R5
037224  010503                          MOV     R5,R3
037226  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
037230  105337  037370                  DECB    $OMODE          ;;TYPE THIS DIGIT?
037234  100034                          BPL     7$              ;;BR IF NO
037236  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
037242  001003                          BNE     4$              ;;TEST FOR 0
037244  005737  037372                  TST     DIGITS          ;;SUPPRESS THIS 0?
037250  001404                          BEQ     5$              ;;BR IF YES
```

```
3562  037252  005237  037372          4$:     INC     DIGITS          ;;DON'T SUPPRESS ANYMORE 0'S
3563  037256  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
3564  037262  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
3565  037266  013704  037300                  MOV     9$,     R4      ;;POINT TO ERRBUF TABLE
3566  037272  110324                          MOVB    R3,     (R4)+   ;;PUT THE CHARACTER IN THE TABLE
3567  037274  105014                          CLRB    (R4)            ;;MAKE IT A MINI-ASCII MSG
3568  037276  104401                          TYPE                    ;;GO TO $TYPE
3569  037300  047052                  9$:     ERRBUF                  ;;HERE IS THE LOCATION OF MSG
3570  037302  062737  000002  037300          ADD     #2,     9$      ;;MOVE TO NEXT TABLE LOC
3571  037310  023727  037300  111052          CMP     9$,     #END    ;;AT THE END OF ALLOWED BUF AREA?
3572  037316  002403                          BLT     7$              ;;BRANCH IF NO
3573  037320  012737  047052  037300          MOV     #ERRBUF,9$      ;;ELSE POINT TO BEGINNING AGAIN
3574  037326  105337  037366          7$:     DECB    $OCNT           ;;COUNT BY 1
3575  037332  003331                          BGT     2$              ;;BR IF MORE TO DO
3576  037334  002403                          BLT     6$              ;;BR IF DONE
3577  037336  005237  037372                  INC     DIGITS          ;;INSURE LAST DIGIT ISN'T A BLANK
3578  037342  000725                          BR      2$              ;;GO DO THE LAST DIGIT
3579  037344  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
3580  037346  012604                          MOV     (SP)+,R4        ;;RESTORE R4
3581  037350  012603                          MOV     (SP)+,R3        ;;RESTORE R3
3582  037352  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
3583  037360  012616                          MOV     (SP)+,(SP)
3584  037362  000002                          RTI                     ;;RETURN
3585  037364     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
3586  037365     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
3587  037366     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
3588  037367     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
3589  037370  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
3590  037372  000000          DIGITS: .WORD   0
```

```
                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

                                ;;**********************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                ;*REPLACED WITH SPACES.
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                ;*      TYPDS                   ;;GO TO THE ROUTINE

037374                          $TYPDS:
037374  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
037376  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
037400  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
037402  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
037404  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
037406  012746  020200                 MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
037412  016605  000020                 MOV     20(SP),R5       ;;GET THE INPUT NUMBER
037416  100004                         BPL     1$              ;;BR IF INPUT IS POS.
037420  005405                         NEG     R5              ;;MAKE THE BINARY NUMBER POS.
037422  112766  000055  000001         MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
037430  005000                 1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
037432  012703  037614                 MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
037436  112723  000040                 MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
037442  005002                 2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
037444  016001  037604                 MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
037450  160105                 3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
037452  002402                         BLT     4$              ;;BR IF DONE
037454  005202                         INC     R2              ;;INCREASE THE BCD DIGIT BY 1
037456  000774                         BR      3$
037460  060105                 4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
037462  005702                         TST     R2              ;;CHECK IF BCD DIGIT=0
037464  001002                         BNE     5$              ;;FALL THROUGH IF 0
037466  105716                         TSTB    (SP)            ;;STILL DOING LEADING 0'S?
037470  100407                         BMI     7$              ;;BR IF YES
037472  106316                 5$:     ASLB    (SP)            ;;MSD?
037474  103003                         BCC     6$              ;;BR IF NO
037476  116663  000001  177777         MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
037504  052702  000060         6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
037510  052702  000040         7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
037514  110223                         MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
037516  005720                         TST     (R0)+           ;;JUST INCREMENTING
037520  020027  000010                 CMP     R0,#10          ;;CHECK THE TABLE INDEX
037524  002746                         BLT     2$              ;;GO DO THE NEXT DIGIT
037526  003002                         BGT     8$              ;;GO TO EXIT
037530  010502                         MOV     R5,R2           ;;GET THE LSD
037532  000764                         BR      6$              ;;GO CHANGE TO ASCII
037534  105726                 8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
037536  100003                         BPL     9$              ;;BR IF NO
037540  116663  177777  177776         MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
037546  105013                 9$:     CLRB    (R3)            ;;SET THE TERMINATOR
037550  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
037552  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
037554  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
3647  037556  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
3648  037560  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
3649  037562  000240                        NOP
3650  037564  104401  037614                TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
3651  037570  000240                        NOP
3652  037572  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
3653  037600  012616                        MOV     (SP)+,(SP)
3654  037602  000002                        RTI                     ;;RETURN TO USER
3655  037604  023420                $DTBL:  10000.
3656  037606  001750                        1000.
3657  037610  000144                        100.
3658  037612  000012                        10.
3659  037614  000004                $DBLK:  .BLKW   4
```

```
3660                                    .SBTTL  TYPE SERVICE
3661  037624                            $TYPE:
3662  037624  132737  000040  014357            BITB    #40,    $ENVM           ;INHIBIT PRINT OUT?
3663  037632  001403                            BEQ     6$              ;BRANCH IF NO
3664  037634  062716  000002                    ADD     #2,     (SP)    ;SET RETURN
3665  037640  000002                            RTI
3666  037642                            6$:
3667  037642  005737  016774                    TST     ERRLCK                  ;IS A CPU IN THE ERROR ROUTINE?
3668  037646  001003                            BNE     3$                      ;BRANCH IF NO
3669  037650  020037  017056                    CMP     R0,     TYPLCK          ;IS THIS CPU FROM THE ERROR ROUTINE?
3670  037654  001363                            BNE     $TYPE                   ;BRANCH IF NO
3671  037656  010246                    3$:     MOV     R2,     -(SP)           ;STORE REGISTERS USED IN THIS PROGRAM
3672  037660  010346                            MOV     R3,     -(SP)
3673  037662  010446                            MOV     R4,     -(SP)
3674  037664  010546                            MOV     R5,     -(SP)
3675  037666  105737  016737                    TSTB    MPF
3676  037672  001003                            BNE     1$
3677  037674  012704  000400                    MOV     #400,   R4      ;DONT TYPE A CPUID
3678  037700  000414                            BR      2$
3679  037702  017705  157114            1$:     MOV     @ACR,   R5              ;PUT SELF ID INTO R5
3680  037706  072527  177770                    ASH     #-10,   R5
3681  037712  010504                            MOV     R5,     R4              ;PUT ID INTO R4
3682  037714  016002  017004                    MOV     NOPRMP(R0),R2   ;COPY THE PROMPT FLAG
3683  037720  000302                            SWAB    R2                      ;GET IT INTO THE LEFT HALF
3684  037722  050204                            BIS     R2,R4                   ;PUT IT IN R5
3685  037724  012760  000001  017004            MOV     #1,     NOPRMP(R0)
3686  037732  106237  017016            2$:     ASRB    TQL1                    ;ENTER TYPQUE CRITICAL SECTION
3687  037736  103375                            BCC     2$
3688                                            ;;******** ENQUE *********
3689  037740  013703  014720                    MOV     TYPQUE+2,R3     ;COPY REAR INDEX
3690  037744  005723                            TST     (R3)+                   ;INCREMENT BY 2
3691  037746  010413                    64$:    MOV     R4,(R3) ;QUEUE THE ELEMENT
3692  037750  010337  014720                    MOV     R3,TYPQUE+2     ;UPDATE THE REAR INDEX
3693                                            ;;*************************
3694  037754                            4$:
3695                                            ;;******** ENQUE *********
3696  037754  013703  014720                    MOV     TYPQUE+2,R3     ;COPY REAR INDEX
3697  037760  005723                            TST     (R3)+                   ;INCREMENT BY 2
3698  037762  016613  000010            65$:    MOV     10(SP),(R3)     ;QUEUE THE ELEMENT
3699  037766  010337  014720                    MOV     R3,TYPQUE+2     ;UPDATE THE REAR INDEX
3700                                            ;;*************************
3701  037772  013343                            MOV     @(R3)+, -(R3)   ;GET PC OF MSG
3702  037774  012737  000001  017016    5$:     MOV     #1,TQL1                 ;CLEAR CRITICAL SECTION
3703  040002  005700                            TST     R0                      ;IS THIS THE MASTER?
3704  040004  001407                            BEQ     11$                     ;BRANCH IF YES.
3705  040006  012605                            MOV     (SP)+,  R5      ;RESTORE THE REGISTERS
3706  040010  012604                            MOV     (SP)+,  R4
3707  040012  012603                            MOV     (SP)+,  R3
3708  040014  012602                            MOV     (SP)+,  R2
3709  040016  062716  000002                    ADD     #2,     (SP)
3710  040022  000002                            RTI                             ;RETURN
3711  040024                            10$:
3712  040024  106237  017016            11$:    ASRB    TQL1                    ;ENTER CRITICAL SECTOR
3713  040030  103375                            BCC     11$
3714  040032                            HERE:
3715                                            ;;****** DEQUE *********
```

```
3716  040032  023737  014716  014720          CMP     TYPQUE,TYPQUE+2 ;;ARE THE INDICIES EQUAL?
3717  040040  001456                          BEQ     13$             ;;YES,THE QUEUE IS EMPTY
3718  040042  013703  014716                  MOV     TYPQUE,R3               ;;COPY FRONT INDEX INTO R3
3719  040046  005723                          TST     (R3)+           ;;INC. R3 BY TWO
3720  040050  011304                          MOV     (R3),R4 ;;DEQUEUE AN ELEMENT
3721  040052  010337  014716                  MOV     R3,TYPQUE               ;;UPDATE FRONT INDEX
3722                                          ;;**********************
3723  040056  032704  000400                  BIT     #BIT8,R4        ;WAS THE NOPRMP(R0) BIT SET?
3724  040062  001025                          BNE     12$
3725  040064  062704  021060                  ADD     #'0',R4         ;MAKE CPUID A CHARACTER
3726  040070  112702  000015                  MOVB    #CR,R2          ;TYPE # CARRAGE RETURN
3727  040074  004737  040236                  JSR     PC,TYPIT        ;TYPE CRLF
3728  040100  112702  000012                  MOVB    #LF,R2
3729  040104  004737  040236                  JSR     PC,TYPIT        ;TYPE IT
3730  040110  110402                          MOVB    R4,R2           ;CPU ID
3731  040112  004737  040236                  JSR     PC,TYPIT
3732  040116  112702  021076                  MOVB    #'>',R2         ; '>'
3733  040122  004737  040236                  JSR     PC,TYPIT        ;TYPE IT
3734  040126  112702  021040                  MOVB    #' ',R2
3735  040132  004737  040236                  JSR     PC,TYPIT        ;TYPE A SPACE
3736  040136                          12$:
3737                                          ;;****** DEQUE **********
3738  040136  023737  014716  014720          CMP     TYPQUE,TYPQUE+2 ;;ARE THE INDICIES EQUAL?
3739  040144  001414                          BEQ     13$             ;;YES,THE QUEUE IS EMPTY
3740  040146  013703  014716                  MOV     TYPQUE,R3               ;;COPY FRONT INDEX INTO R3
3741  040152  005723                          TST     (R3)+           ;;INC. R3 BY TWO
3742  040154  011304                          MOV     (R3),R4 ;;DEQUEUE AN ELEMENT
3743  040156  010337  014716                  MOV     R3,TYPQUE               ;;UPDATE FRONT INDEX
3744                                          ;;**********************
3745  040162  112402                  14$:    MOVB    (R4)+,R2        ;GET A CHARECTER
3746  040164  005702                          TST     R2
3747  040166  001721                          BEQ     HERE            ;WE'RE DONE ,R2 IS CLEAR
3748  040170  004737  040236                  JSR     PC,TYPIT        ;TYPE THE CHARECTER
3749  040174  000772                          BR      14$             ;LOOP
```

```
 3750  040176  012737  014722  014716  13$:    MOV     #TYPQUE+4,      TYPQUE
 3751  040204  012737  014722  014720          MOV     #TYPQUE+4,      TYPQUE+2
 3752  040212  012737  000001  017016          MOV     #1,     TQL1                    ;RETURN
 3753  040220  012605                          MOV     (SP)+,  R5      ;RESTORE THE REGISTERS
 3754  040222  012604                          MOV     (SP)+,  R4
 3755  040224  012603                          MOV     (SP)+,  R3
 3756  040226  012602                          MOV     (SP)+,  R2
 3757  040230  062716  000002                  ADD     #2,     (SP)
 3758  040234  000002                          RTI
 3759
 3760  040236  105777  153752          TYPIT:  TSTB    @$TPS           ;WAIT UNTIL PRINTER IS READY
 3761  040242  100375                          BPL     TYPIT
```

```
3762  040244  110277  153746          MOVB    R2,@$TPB          ;TYPE THE CHARACTER
3763  040250  000207                  RTS     PC                ;RETURN
```

```
                                        .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.

040252  010260  014552        $TRAP:  MOV     R2,     SAVRG(R0)
040256  011602                1$:     MOV     (SP),R2         ;;GET TRAP ADDRESS
040260  005742                        TST     -(R2)           ;;BACKUP BY 2
040262  111202                        MOVB    (R2),R2         ;;GET RIGHT BYTE OF TRAP
040264  006302                        ASL     R2              ;;POSITION FOR INDEXING
040266  016202  040320                MOV     $TRPAD(R2),R2   ;;INDEX TO TABLE
040272  010260  014562                MOV     R2,     ROUTE(R0)
040276  016002  014552                MOV     SAVRG(R0),      R2
040302  000170  014562                JMP     @ROUTE(R0)


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

040306  011646                $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
040310  016666  000004  000002        MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
040316  000002                        RTI                     ;;RESTORE THE PSW

                                        .SBTTL  TRAP TABLE

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

                                        ;       ROUTINE
                                        ;       -------
040320  040306                $TRPAD: .WORD   $TRAP2
040322  037624                        $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
040324  037120                        $TYPOC  ;;CALL-TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
040326  037074                        $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
040330  037134                        $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
040332  037374                        $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)


040334  000170  014672        ISTDIS: JMP     @ISTTAB(R0)     ;IIST INTERRUPT DISPATCHER
```

```
 3808  040340  012737  177777  014602  PWRDIS: MOV     #-1,FLAG            ;FIRST INSTRUCTION FLAG
 3809  040346  005037  014602                   CLR     FLAG                ;NOW CLEAR IT
 3810  040352  000177  154304                   JMP     @PWRTAB
 3811  040356  000170  014702          ERRDIS: JMP     @ERRTAB(R0)         ;CPU ERROR DISPATCHER
 3812
 3813  040362                          CPUER:
 3814  040362  013760  177766  014250          MOV     @#CPUERR,$REG1(R0)          ;CPU ERROR REG.
 3815  040370  104005                          ERROR   5
 3816  040372  000000                          HALT
 3817  040374  000002                          RTI                         ;RETURN
 3818                                          .SBTTL  DATA AREA
 3819  040376  005015  042503  041113  TM1:    .ASCIZ  <CR><LF>\CEKBG-B  11/70-74MP SYSTEM POWER FAIL DIAGNOSTIC\
 3820  040404  026507  020102  030440
 3821  040412  027461  030067  033455
 3822  040420  046464  020120  054523
 3823  040426  052123  046505  050040
 3824  040434  053517  051105  043040
 3825  040442  044501  020114  044504
 3826  040450  043501  047516  052123
 3827  040456  041511     000
 3828  040461     015  051412  044527  TM2:    .ASCIZ  <CR><LF>\SWITCH REGISTER = \
 3829  040466  041524  020110  042522
 3830  040474  044507  052123  051105
 3831  040502  036440  000040
 3832  040506  005015  052533  044516  TM4:    .ASCIZ  <CR><LF /[UNIBUS EXERCISER WILL BE USED]/
 3833  040514  052502  020123  054105
 3834  040522  051105  044503  042523
 3835  040530  020122  044527  046114
 3836  040536  041040  020105  051525
 3837  040544  042105  000135
 3838  040550  005015  052533  044516  TM5:    .ASCIZ  <CR><LF>/[UNIBUS EXERCISER WILL NOT BE USED]/
 3839  040556  052502  020123  054105
 3840  040564  051105  044503  042523
 3841  040572  020122  044527  046114
 3842  040600  047040  052117  041040
 3843  040606  020105  051525  042105
 3844  040614  000135
 3845  040616  005015  046533  046125  TM6:    .ASCIZ  <CR><LF>/[MULTIPROCESSOR MODE IS IN EFFECT]/
 3846  040624  044524  051120  041517
 3847  040632  051505  047523  020122
 3848  040640  047515  042504  044440
 3849  040646  020123  047111  042440
 3850  040654  043106  041505  056524
 3851  040662     000
 3852  040663     015  055412  047125  TM7:    .ASCIZ  <CR><LF>/[UNIPROCESSOR MODE IS IN EFFECT]/
 3853  040670  050111  047522  042503
 3854  040676  051523  051117  046440
 3855  040704  042117  020105  051511
 3856  040712  044440  020116  043105
 3857  040720  042506  052103  000135
 3858  040726  055440  052117  042510  TM10:   .ASCIZ  / [OTHER CPUS ARE RUNNING.]/
 3859  040734  020122  050103  051525
 3860  040742  040440  042522  051040
 3861  040750  047125  044516  043516
 3862  040756  056456     000
 3863  040761     015  003412  050103  TM11:   .ASCIZ  <CR><LF><07>/CPU #/
```

```
 3864  040766  020125  000043
 3865  040772  051440  042520  044503  TM12:   .ASCIZ  / SPECIFIED BUT NOT ACTIVE/
 3866  041000  044506  042105  041040
 3867  041006  052125  047040  052117
 3868  041014  040440  052103  053111
 3869  041022  000105
```

```
3870  041024  005015  047111  042524  TM13:   .ASCII  <CR><LF>/INTERRUPT THE POWER AFTER THE TEST NUMBER APPEARS/
3871  041032  051122  050125  020124
3872  041040  044124  020105  047520
3873  041046  042527  020122  043101
3874  041054  042524  020122  044124
3875  041062  020105  042524  052123
3876  041070  047040  046525  042502
3877  041076  020122  050101  042520
3878  041104  051101     123
3879  041107     040  047111  052040          .ASCII  / IN THE DISPLAY./<CR><LF>
3880  041114  042510  042040  051511
3881  041122  046120  054501  006456
3882  041130     012
3883  041131     111  020106  047531          .ASCII  /IF YOU HAVE AN RD CONSOLE, INTERRUPT THE POWER/
3884  041136  020125  040510  042526
3885  041144  040440  020116  042122
3886  041152  041440  047117  047523
3887  041160  042514  020054  047111
3888  041166  042524  051122  050125
3889  041174  020124  044124  020105
3890  041202  047520  042527     122
3891  041207     015  040412  020124          .ASCII  <CR><LF>/AT THE END OF THIS MESSAGE.  THEREAFTER,INTERRUPT THE POWER 15(
3892  041214  044124  020105  047105
3893  041222  020104  043117  052040
3894  041230  044510  020123  042515
3895  041236  051523  043501  027105
3896  041244  020040  044124  051105
3897  041252  040505  052106  051105
3898  041260  044454  052116  051105
3899  041266  052522  052120  052040
3900  041274  042510  050040  053517
3901  041302  051105  030440  024065
3902  041310  024470  046440  051117
3903  041316  020105  044524  042515
3904  041324     123
3905  041325     015  052012  020117          .ASCIZ  <CR><LF>/TO REACH THE END OF SECTION 1./<CR><LF>
3906  041332  042522  041501  020110
3907  041340  044124  020105  047105
3908  041346  020104  043117  051440
3909  041354  041505  044524  047117
3910  041362  030440  006456  000012
3911  041370  005015  042412  052116  TM14:   .ASCIZ<CR><LF><LF>/ENTERING SECTION 1/<CR><LF>
3912  041376  051105  047111  020107
3913  041404  042523  052103  047511
3914  041412  020116  006461  000012
3915  041420  005015  042412  052116  TM15:   .ASCIZ<CR><LF><LF>/ENTERING SECTION 2/<CR><LF>
3916  041426  051105  047111  020107
3917  041434  042523  052103  047511
3918  041442  020116  006462  000012
3919  041450  006412  044124  020105  TM76:   .ASCIZ<12><15>/THE MASTER IS CPU #/
3920  041456  040515  052123  051105
3921  041464  044440  020123  050103
3922  041472  020125  000043
3923  041476  005012  052015  051505  TM77:   .ASCIZ<12><12><15>/TEST /
3924  041504  020124     000
3925  041507     015  050012  053517  TM100:  .ASCIZ  <CR><LF>/POWER FAIL CPU # /
```

```
3926  041514  051105  043040  044501
3927  041522  020114  050103  020125
3928  041530  020043     000
3929  041533     015  005012  047516  TM101:  .ASCIZ  <CR><LF><LF>/NO MASSBUS DEVICE AVAILABLE ON CPU # /
3930  041540  046440  051501  041123
3931  041546  051525  042040  053105
3932  041554  041511  020105  053101
3933  041562  044501  040514  046102
3934  041570  020105  047117  041440
3935  041576  052520  021440  000040
3936  041604  005015  051120  041517  TM102:  .ASCIZ  <CR><LF>/PROCEEDING TO NEXT CPU/
3937  041612  042505  044504  043516
3938  041620  052040  020117  042516
3939  041626  052130  041440  052520
3940  041634     000
3941  041635     015  050012  053517  TM103:  .ASCIZ  <CR><LF>/POWER FAILURE ON CPU AS EXPECTED/
3942  041642  051105  043040  044501
3943  041650  052514  042522  047440
3944  041656  020116  050103  020125
3945  041664  051501  042440  050130
3946  041672  041505  042524  000104
3947  041700  005015  042507  020124  TM104:  .ASCII  <CR><LF>/GET SET TO POWER FAIL ENTIRE SYSTEM.../
3948  041706  042523  020124  047524
3949  041714  050040  053517  051105
3950  041722  043040  044501  020114
3951  041730  047105  044524  042522
3952  041736  051440  051531  042524
3953  041744  027115  027056
3954  041750  005012  050015  052125          .ASCII<12><12><15>/PUT BATTERY BACKUP ON ALL MEM BOXES/
3955  041756  041040  052101  042524
3956  041764  054522  041040  041501
3957  041772  052513  020120  047117
3958  042000  040440  046114  046440
3959  042006  046505  041040  054117
3960  042014  051505
3961  042016  006412  040515  042513          .ASCII<12><15>/MAKE ALL MEMORY PORTS ONLINE/
3962  042024  040440  046114  046440
3963  042032  046505  051117  020131
3964  042040  047520  052122  020123
3965  042046  047117  044514  042516
3966  042054  006412  040515  042513          .ASCII<12><15>/MAKE ALL CPU POWER-UP SWITCHES 'RUN OR HALT''/
3967  042062  040440  046114  041440
3968  042070  052520  050040  053517
3969  042076  051105  052455  020120
3970  042104  053523  052111  044103
3971  042112  051505  021040  052522
3972  042120  020116  051117  044040
3973  042126  046101  021124
3974  042132  005012  052015  042510          .ASCII<12><12><15>/THEN POWER FAIL THE ENTIRE SYSTEM/
3975  042140  020116  047520  042527
3976  042146  020122  040506  046111
3977  042154  052040  042510  042440
3978  042162  052116  051111  020105
3979  042170  054523  052123  046505
3980  042176  005012  051015  051505          .ASCII<12><12><15>/RESTORE POWER 5 SECONDS AFTER POWER FAIL/
3981  042204  047524  042522  050040
```

```
 3982  042212  053517  051105  032440
 3983  042220  051440  041505  047117
 3984  042226  051504  040440  052106
 3985  042234  051105  050040  053517
 3986  042242  051105  043040  044501
 3987  042250     114
 3988  042251     012  042415  041501          .ASCIZ<12><15>/EACH CPU SHOULD REPORT A POWER FAIL/
 3989  042256  020110  050103  020125
 3990  042264  044123  052517  042114
 3991  042272  051040  050105  051117
 3992  042300  020124  020101  047520
 3993  042306  042527  020122  040506
 3994  042314  046111     000
 3995  042317     015  005012  042507  TM106:  .ASCIZ  <CR><LF><LF>/GET SET TO POWER FAIL MEM BOX # /
 3996  042324  020124  042523  020124
 3997  042332  047524  050040  053517
 3998  042340  051105  043040  044501
 3999  042346  020114  042515  020115
 4000  042354  047502  020130  020043
 4001  042362     000
 4002  042363     012  006412  052520  TM107:  .ASCII<12><12><15> /PUT BATTERY BACKUP ON ALL MEMORY BOXES/
 4003  042370  020124  040502  052124
 4004  042376  051105  020131  040502
 4005  042404  045503  050125  047440
 4006  042412  020116  046101  020114
 4007  042420  042515  047515  054522
 4008  042426  041040  054117  051505
 4009  042434  006412  040515  042513          .ASCII<12><15>/MAKE ALL MEMORY PORTS ONLINE/
 4010  042442  040440  046114  046440
 4011  042450  046505  051117  020131
 4012  042456  047520  052122  020123
 4013  042464  047117  044514  042516
 4014  042472  006412  040515  042513          .ASCII<12><15>/MAKE ALL CPU POWER-UP SWITCHES 'RUN OR HALT''/
 4015  042500  040440  046114  041440
 4016  042506  052520  050040  053517
 4017  042514  051105  052455  020120
 4018  042522  053523  052111  044103
 4019  042530  051505  021040  052522
 4020  042536  020116  051117  044040
 4021  042544  046101  021124
 4022  042550  005012  052015  042510          .ASCII<12><12><15>/THEN POWER FAIL THE MEM BOX/
 4023  042556  020116  047520  042527
 4024  042564  020122  040506  046111
 4025  042572  052040  042510  046440
 4026  042600  046505  041040  054117
 4027  042606  005012  051015  051505          .ASCII<12><12><15>/RESTORE POWER 5 SECONDS AFTER POWER FAIL/
 4028  042614  047524  042522  050040
 4029  042622  053517  051105  032440
 4030  042630  051440  041505  047117
 4031  042636  051504  040440  052106
 4032  042644  051105  050040  053517
 4033  042652  051105  043040  044501
 4034  042660     114
 4035  042661     012  052015  042510          .ASCII<12><15>/THEN TYPE ANY CHARACTER AT THE MASTER CONSOLE/
 4036  042666  020116  054524  042520
 4037  042674  040440  054516  041440
```

```
 4038  042702  040510  040522  052103
 4039  042710  051105  040440  020124
 4040  042716  044124  020105  040515
 4041  042724  052123  051105  041440
 4042  042732  047117  047523  042514
 4043  042740  006412  040505  044103          .ASCIZ<12><15>/EACH CPU SHOULD REPORT A POWER FAILURE/
 4044  042746  041440  052520  051440
 4045  042754  047510  046125  020104
 4046  042762  042522  047520  052122
 4047  042770  040440  050040  053517
 4048  042776  051105  043040  044501
 4049  043004  052514  042522     000
 4050  043011     012  006412  052520  TM108:  .ASCII<12><12><15>/PUT BATTERY BACKUP ON ALL MEMORY BOXES/
 4051  043016  020124  040502  052124
 4052  043024  051105  020131  040502
 4053  043032  045503  050125  047440
 4054  043040  020116  046101  020114
 4055  043046  042515  047515  054522
 4056  043054  041040  054117  051505
 4057  043062  006412  040515  042513          .ASCII<12><15>/MAKE ALL MEMORY PORTS OFFLINE ON MEM BOX TO BE POWER-FAILED/
 4058  043070  040440  046114  046440
 4059  043076  046505  051117  020131
 4060  043104  047520  052122  020123
 4061  043112  043117  046106  047111
 4062  043120  020105  047117  046440
 4063  043126  046505  041040  054117
 4064  043134  052040  020117  042502
 4065  043142  050040  053517  051105
 4066  043150  043055  044501  042514
 4067  043156     104
 4068  043157     012  046415  045501          .ASCII<12><15>/MAKE ALL CPU POWER-UP SWITCHES 'RUN OR HALT''/
 4069  043164  020105  046101  020114
 4070  043172  050103  020125  047520
 4071  043200  042527  026522  050125
 4072  043206  051440  044527  041524
 4073  043214  042510  020123  051042
 4074  043222  047125  047440  020122
 4075  043230  040510  052114     042
 4076  043235     012  006412  044124          .ASCII<12><12><15>/THEN POWER FAIL THE MEM BOX/
 4077  043242  047105  050040  053517
 4078  043250  051105  043040  044501
 4079  043256  020114  044124  020105
 4080  043264  042515  020115  047502
 4081  043272     130
 4082  043273     012  006412  042522          .ASCII<12><12><15>/RESTORE POWER 5 SECONDS AFTER POWER FAIL/
 4083  043300  052123  051117  020105
 4084  043306  047520  042527  020122
 4085  043314  020065  042523  047503
 4086  043322  042116  020123  043101
 4087  043330  042524  020122  047520
 4088  043336  042527  020122  040506
 4089  043344  046111
 4090  043346  006412  042522  052123          .ASCII<12><15>/RESTORE ALL MEMORY PORTS ONLINE/
 4091  043354  051117  020105  046101
 4092  043362  020114  042515  047515
 4093  043370  054522  050040  051117
```

```
 4094  043376  051524  047440  046116
 4095  043404  047111     105
 4096  043407     012  052015  042510          .ASCII<12><15>/THEN TYPE ANY CHARACTER AT THE MASTER CONSOLE/
 4097  043414  020116  054524  042520
 4098  043422  040440  054516  041440
 4099  043430  040510  040522  052103
 4100  043436  051105  040440  020124
 4101  043444  044124  020105  040515
 4102  043452  052123  051105  041440
 4103  043460  047117  047523  042514
 4104  043466  006412  047516  041440          .ASCIZ<12><15>/NO CPU SHOULD REPORT A POWER FAILURE/
 4105  043474  052520  051440  047510
 4106  043502  046125  020104  042522
 4107  043510  047520  052122  040440
 4108  043516  050040  053517  051105
 4109  043524  043040  044501  052514
 4110  043532  042522     000
 4111  043535     012  005015  040527  $PGM1:  .ASCIZ <12><15><12>/WARNING:  DRIVE # /
 4112  043542  047122  047111  035107
 4113  043550  020040  051104  053111
 4114  043556  020105  020043     000
 4115  043563     012  047415  020116  $PGM2:  .ASCIZ<12><15> /ON CPU # /
 4116  043570  050103  020125  020043
 4117  043576     000
 4118  043577     012  044415  020123  $PGM3:  .ASCII <12><15>/IS PROGRAMMABLE OVER BOTH A AND B PORTS/
 4119  043604  051120  043517  040522
 4120  043612  046515  041101  042514
 4121  043620  047440  042526  020122
 4122  043626  047502  044124  040440
 4123  043634  040440  042116  041040
 4124  043642  050040  051117  051524
 4125  043650  006412  044124  020105          .ASCIZ<12><15>/THE DRIVE WILL BE USED LATER IN THIS DIAGNOSTIC/
 4126  043656  051104  053111  020105
 4127  043664  044527  046114  041040
 4128  043672  020105  051525  042105
 4129  043700  046040  052101  051105
 4130  043706  044440  020116  044124
 4131  043714  051511  042040  040511
 4132  043722  047107  051517  044524
 4133  043730  000103
 4134  043732  006412  047516  046440  NODEV:  .ASCIZ <12><15>/NO MASSBUS DEVICES ON CPU # /
 4135  043740  051501  041123  051525
 4136  043746  042040  053105  041511
 4137  043754  051505  047440  020116
 4138  043762  050103  020125  020043
 4139  043770     000
 4140  043771     012  006412  047117  TM109:  .ASCIZ <12><12><15>/ONLY ONE MEMORY BOX ONLINE- SKIPPING THIS TEST/
 4141  043776  054514  047440  042516
 4142  044004  046440  046505  051117
 4143  044012  020131  047502  020130
 4144  044020  047117  044514  042516
 4145  044026  020055  045523  050111
 4146  044034  044520  043516  052040
 4147  044042  044510  020123  042524
 4148  044050  052123     000
 4149  044053     012  006412  044504  TM110:  .ASCII<12><12><15>/DISABLE BATTERY BACKUP ON MEM BOX TO BE POWER-FAILED/
```

```
 4150  044060  040523  046102  020105
 4151  044066  040502  052124  051105
 4152  044074  020131  040502  045503
 4153  044102  050125  047440  020116
 4154  044110  042515  020115  047502
 4155  044116  020130  047524  041040
 4156  044124  020105  047520  042527
 4157  044132  026522  040506  046111
 4158  044140  042105
 4159  044142  006412  052520  020124          .ASCII<12><15>/PUT ALL SLAVE CPU MEM PORTS ONLINE/
 4160  044150  046101  020114  046123
 4161  044156  053101  020105  050103
 4162  044164  020125  042515  020115
 4163  044172  047520  052122  020123
 4164  044200  047117  044514  042516
 4165  044206  006412  040515  042513          .ASCII<12><15>/MAKE MASTER CPU MEM PORT OFFLINE ON BOX TO BE POWER-FAILED/
 4166  044214  046440  051501  042524
 4167  044222  020122  050103  020125
 4168  044230  042515  020115  047520
 4169  044236  052122  047440  043106
 4170  044244  044514  042516  047440
 4171  044252  020116  047502  020130
 4172  044260  047524  041040  020105
 4173  044266  047520  042527  026522
 4174  044274  040506  046111  042105
 4175  044302  006412  040515  042513          .ASCII<12><15>/MAKE ALL CPU POWER-UP SWITCHES 'RUN OR BOOT''/
 4176  044310  040440  046114  041440
 4177  044316  052520  050040  053517
 4178  044324  051105  052455  020120
 4179  044332  053523  052111  044103
 4180  044340  051505  021040  052522
 4181  044346  020116  051117  041040
 4182  044354  047517  021124
 4183  044360  005012  052015  042510          .ASCII<12><12><15>/THEN POWER FAIL THE MEM BOX/
 4184  044366  020116  047520  042527
 4185  044374  020122  040506  046111
 4186  044402  052040  042510  046440
 4187  044410  046505  041040  054117
 4188  044416  005012  051015  051505          .ASCII<12><12><15>/RESTORE POWER 5 SECONDS AFTER POWER FAIL/
 4189  044424  047524  042522  050040
 4190  044432  053517  051105  032440
 4191  044440  051440  041505  047117
 4192  044446  051504  040440  052106
 4193  044454  051105  050040  053517
 4194  044462  051105  043040  044501
 4195  044470     114
 4196  044471     012  051015  051505          .ASCII<12><15>/RESTORE ALL MEM PORTS ONLINE/
 4197  044476  047524  042522  040440
 4198  044504  046114  046440  046505
 4199  044512  050040  051117  051524
 4200  044520  047440  046116  047111
 4201  044526     105
 4202  044527     012  051015  051505          .ASCII<12><15>/RESTORE ALL CPU POWER-UP SWITCHES TO 'RUN OR HALT''/
 4203  044534  047524  042522  040440
 4204  044542  046114  041440  052520
 4205  044550  050040  053517  051105
```

```
 4206  044556  052455  020120  053523
 4207  044564  052111  044103  051505
 4208  044572  052040  020117  051042
 4209  044600  047125  047440  020122
 4210  044606  040510  052114     042
 4211  044613     012  052015  042510          .ASCII<12><15>/THEN TYPE ANY CHARACTER AT THE MASTER CONSOLE/
 4212  044620  020116  054524  042520
 4213  044626  040440  054516  041440
 4214  044634  040510  040522  052103
 4215  044642  051105  040440  020124
 4216  044650  044124  020105  040515
 4217  044656  052123  051105  041440
 4218  044664  047117  047523  042514
 4219  044672  006412  040505  044103          .ASCIZ<12><15>/EACH SLAVE CPU SHOULD REPORT AN INTERRUPT/
 4220  044700  051440  040514  042526
 4221  044706  041440  052520  051440
 4222  044714  047510  046125  020104
 4223  044722  042522  047520  052122
 4224  044730  040440  020116  047111
 4225  044736  042524  051122  050125
 4226  044744  000124
 4227  044746  006412  050103  020125  TM111:  .ASCIZ<12><15>/CPU INTERRUPTED AS EXPECTED/
 4228  044754  047111  042524  051122
 4229  044762  050125  042524  020104
 4230  044770  051501  042440  050130
 4231  044776  041505  042524  000104
 4232  045004  006412  047520  042527  $DOWN:  .ASCIZ <12><15>/POWER DOWN TIME WAS UNDER 2 MILISECONDS /
 4233  045012  020122  047504  047127
 4234  045020  052040  046511  020105
 4235  045026  040527  020123  047125
 4236  045034  042504  020122  020062
 4237  045042  044515  044514  042523
 4238  045050  047503  042116  020123
 4239  045056     000
 4240  045057     015  052412  042516  EM1:    .ASCIZ  <CR><LF>/UNEXPECTED POWER FAILURE ON CPU/
 4241  045064  050130  041505  042524
 4242  045072  020104  047520  042527
 4243  045100  020122  040506  046111
 4244  045106  051125  020105  047117
 4245  045114  041440  052520     000
 4246  045121     015  052412  042516  EM2:    .ASCIZ  <CR><LF>/UNEXPECTED POWER UP SEQUENCE ON CPU/
 4247  045126  050130  041505  042524
 4248  045134  020104  047520  042527
 4249  045142  020122  050125  051440
 4250  045150  050505  042525  041516
 4251  045156  020105  047117  041440
 4252  045164  052520     000
 4253  045167     015  044412  046114  EM3:    .ASCIZ  <CR><LF>/ILLEGAL POWER DOWN SEQUENCE ON CPU/
 4254  045174  043505  046101  050040
 4255  045202  053517  051105  042040
 4256  045210  053517  020116  042523
 4257  045216  052521  047105  042503
 4258  045224  047440  020116  050103
 4259  045232  000125
 4260  045234  005015  046111  042514  EM4:    .ASCIZ  <CR><LF>/ILLEGAL POWER UP SEQUENCE/
 4261  045242  040507  020114  047520
```

```
 4262  045250  042527  020122  050125
 4263  045256  051440  050505  042525
 4264  045264  041516  000105
 4265  045270  005015  047125  054105  EM5:    .ASCIZ  <CR><LF>/UNEXPECTED TRAP TO 4/
 4266  045276  042520  052103  042105
 4267  045304  052040  040522  020120
 4268  045312  047524  032040     000
 4269  045317     015  052412  042516  EM6:    .ASCIZ  <CR><LF>/UNEXPECTED TRAP TO 10/
 4270  045324  050130  041505  042524
 4271  045332  020104  051124  050101
 4272  045340  052040  020117  030061
 4273  045346     000
 4274  045347     015  052412  042516  EM7:    .ASCIZ  <CR><LF>/UNEXPECTED TRAP TO 114/
 4275  045354  050130  041505  042524
 4276  045362  020104  051124  050101
 4277  045370  052040  020117  030461
 4278  045376  000064
 4279  045400  005015  042101  051104  EM10:   .ASCIZ  <CR><LF>/ADDRESS ON STACK IS WRONG/
 4280  045406  051505  020123  047117
 4281  045414  051440  040524  045503
 4282  045422  044440  020123  051127
 4283  045430  047117  000107
 4284  045434  005015  046117  020104  EM11:   .ASCIZ  <CR><LF>/OLD PS IS WRONG/
 4285  045442  051520  044440  020123
 4286  045450  051127  047117  000107
 4287  045456  005015  042117  020104  EM12:   .ASCIZ  <CR><LF>/ODD ADDRESS TRAP FAILED/
 4288  045464  042101  051104  051505
 4289  045472  020123  051124  050101
 4290  045500  043040  044501  042514
 4291  045506  000104
 4292  045510  005015  042515  047515  EM13:   .ASCIZ  <CR><LF>/MEMORY CORRUPTED ON POWER FAIL/
 4293  045516  054522  041440  051117
 4294  045524  052522  052120  042105
 4295  045532  047440  020116  047520
 4296  045540  042527  020122  040506
 4297  045546  046111     000
 4298  045551     015  052012  046511  EM14:   .ASCIZ  <CR><LF>/TIMEOUT TRAP FAILED/
 4299  045556  047505  052125  052040
 4300  045564  040522  020120  040506
 4301  045572  046111  042105     000
 4302  045577     015  050012  053517  EM15:   .ASCIZ  <CR><LF>/POWER FAIL RETURNED TOO SOON/
 4303  045604  051105  043040  044501
 4304  045612  020114  042522  052524
 4305  045620  047122  042105  052040
 4306  045626  047517  051440  047517
 4307  045634  000116
 4308  045636  005015  047516  020124  EM16:   .ASCIZ  <CR><LF>/NOT ENOUGH OR TOO MANY INSTRUCTIONS EXECUTED/
 4309  045644  047105  052517  044107
 4310  045652  047440  020122  047524
 4311  045660  020117  040515  054516
 4312  045666  044440  051516  051124
 4313  045674  041525  044524  047117
 4314  045702  020123  054105  041505
 4315  045710  052125  042105     000
 4316  045715     015  047012  020117  EM17:   .ASCIZ  <CR><LF>/NO MEM. MANG. VIOLATION OR TRAP TO 4/
 4317  045722  042515  027115  046440
```

```
 4318  045730  047101  027107  053040
 4319  045736  047511  040514  044524
 4320  045744  047117  047440  020122
 4321  045752  051124  050101  052040
 4322  045760  020117  000064
 4323  045764  005015  047516  044440  EM20:   .ASCIZ  <CR><LF>/NO IIST INTERRUPT/
 4324  045772  051511  020124  047111
 4325  046000  042524  051122  050125
 4326  046006  000124
 4327  046010  005015  047111  047503  EM21:   .ASCIZ  <CR><LF>?INCORRECT BRK AND/OR DCF FLAGS?
 4328  046016  051122  041505  020124
 4329  046024  051102  020113  047101
 4330  046032  027504  051117  042040
 4331  046040  043103  043040  040514
 4332  046046  051507     000
 4333  046051     015  041412  052520  EM22:   .ASCIZ  <CR><LF>/CPU DID NOT TRAP TO VIRTUAL 24/
 4334  046056  042040  042111  047040
 4335  046064  052117  052040  040522
 4336  046072  020120  047524  053040
 4337  046100  051111  052524  046101
 4338  046106  031040  000064
 4339  046112  005015  044103  041505  EM23:   .ASCIZ  <CR><LF>/CHECKSUM ON MASSBUS TRANSFER IS WRONG/
 4340  046120  051513  046525  047440
 4341  046126  020116  040515  051523
 4342  046134  052502  020123  051124
 4343  046142  047101  043123  051105
 4344  046150  044440  020123  051127
 4345  046156  047117  000107
 4346  046162  006412  047516  050040  EM24:   .ASCIZ <12><15>/NO POWER FAIL ON CPU/
 4347  046170  053517  051105  043040
 4348  046176  044501  020114  047117
 4349  046204  041440  052520     000
 4350  046211     012  052415  042516  EM25:   .ASCIZ <12><15>/UNEXPECTED CPU INTERRUPT/
 4351  046216  050130  041505  042524
 4352  046224  020104  050103  020125
 4353  046232  047111  042524  051122
 4354  046240  050125  000124
 4355
 4356  046244  005015  044511  052123  DH5:    .ASCIZ  <CR><LF>/IISTID   PC       CPUERR/
 4357  046252  042111  020011  050040
 4358  046260  020103  020040  020040
 4359  046266  020040  050103  042525
 4360  046274  051122     000
 4361  046277     015  020012  044440  DH7:    .ASCIZ  <CR><LF>/  IISTID    ERRORPC  CPUERR   MEMERR/
 4362  046304  051511  044524  020104
 4363  046312  020040  042440  051122
 4364  046320  051117  041520  020040
 4365  046326  050103  042525  051122
 4366  046334  020040  046440  046505
 4367  046342  051105  000122
 4368  046346  005015  042524  052123  DH10:   .ASCIZ  <CR><LF>/TESTNO   ERRORPC/
 4369  046354  047516  020040  042440
 4370  046362  051122  051117  041520
 4371  046370     000
 4372  046371     015  052012  051505  DH11:   .ASCIZ  <CR><LF>/TESTNO   ERRORPC     PS/
 4373  046376  047124  020117  020040
```

```
 4374  046404  051105  047522  050122
 4375  046412  020103  020040  020040
 4376  046420  051520     000
 4377  046423     012  052015  051505  DH12:   .ASCIZ<12><15>/TESTNO   ERRORPC   PAGE ADDRESS REGISTER OF BAD MEMORY/
 4378  046430  047124  004517  051105
 4379  046436  047522  050122  020103
 4380  046444  020040  040520  042507
 4381  046452  040440  042104  042522
 4382  046460  051523  051040  043505
 4383  046466  051511  042524  020122
 4384  046474  043117  041040  042101
 4385  046502  046440  046505  051117
 4386  046510  000131
 4387  046512  005015  042524  052123  DH14:   .ASCIZ  <CR><LF>/TESTNO   ERRORPC  CPUERR/
 4388  046520  047516  020040  042440
 4389  046526  051122  051117  041520
 4390  046534  020040  050103  042525
 4391  046542  051122     000
 4392  046545     015  052012  051505  DH20:   .ASCIZ  <CR><LF>/TESTNO    IISTID     ACR     PGTE     PGCS/
 4393  046552  047124  020117  020040
 4394  046560  044440  051511  044524
 4395  046566  020104  020040  020040
 4396  046574  041501  020122  020040
 4397  046602  020040  043520  042524
 4398  046610  020040  020040  050040
 4399  046616  041507  000123
 4400  046622  005015  042524  052123  DH21:   .ASCIZ  <CR><LF>/TESTNO    IISTID   FOUND    SHOULD BE/
 4401  046630  047516  020040  020040
 4402  046636  044511  052123  042111
 4403  046644  020040  043040  052517
 4404  046652  042116  020040  020040
 4405  046660  044123  052517  042114
 4406  046666  041040  000105
 4407  046672  005015  042524  052123  DH22:   .ASCIZ  <CR><LF>/TESTNO   IISTID    ERRORPC/
 4408  046700  047516  020040  044440
 4409  046706  051511  044524  020104
 4410  046714  020040  042440  051122
 4411  046722  051117  041520     000
 4412          046730                        .EVEN
 4413
 4414  046730  014240  014064  014250  DT5:    $REG0,$ERRPC,$REG1,0
 4415  046736  000000
 4416  046740  014240  014064  014250  DT7:    $REG0,$ERRPC,$REG1,$REG2,0
 4417  046746  014260  000000
 4418  046752  014002  014064  000000  DT10:   $TSTNM,$ERRPC,0
 4419  046760  014002  014064  014240  DT11:   $TSTNM,$ERRPC,$REG0,0
 4420  046766  000000
 4421  046770  014002  014064  014250  DT12:   $TSTNM,$ERRPC,$REG1,0
 4422  046776  000000
 4423  047000  014002  014240  014250  DT20:   $TSTNM,$REG0,$REG1,$REG2,$REG3,0
 4424  047006  014260  014270  000000
 4425  047014  014002  014240  014250  DT21:   $TSTNM,$REG0,$REG1,$REG2,0
 4426  047022  014260  000000
 4427  047026  014002  014240  014064  DT22:   $TSTNM,$REG0,$ERRPC,0
 4428  047034  000000
 4429  047036  005015  020133  045517  OK:     .ASCIZ  <CR><LF>/[ OK ]  /
```

```
 4430  047044  056440  020040       000
 4431          047052                              .EVEN
 4432  047052  001000                     ERRBUF: .BLKW   1000           ;ERROR ASCII MSG STORAGE AREA
 4433  051052  020000                     DSKBUF: .BLKW   20000          ;MASSBUS BUFFER AREA
 4434  111052  000000                     END:    0
```

```
4435
4436         000001                 .END
```

```
A       024510          1821    1827#
ABASE = 000000           722     763
ACDW1 = 000000           722     765
ACDW2 = 000000           722     766
ACPUOP= 000000           722     737
ACR     017022           866#    1108*   1109*   1114    1124*   1145*   1151*   1154*   1189    1192*   1193*   1210*   1214*
                        1218*   1226*   1236*   1778*   1784    1868*   1929*   1932*   1937    1938*   1944*   1971*   2040*
                        2197*   2248*   2306*   2355*   2406*   2464*   2512*   2563*   2621*   2805    2816*   3031    3679
ADDW0 = 000000           722     767
ADDW1 = 000000           722     768
ADDW10= 000000           722     777
ADDW11= 000000           722     778
ADDW12= 000000           722     779
ADDW13= 000000           722     780
ADDW14= 000000           722     781
ADDW15= 000000           722     782
ADDW2 = 000000           722     769
ADDW3 = 000000           722     770
ADDW4 = 000000           722     771
ADDW5 = 000000           722     772
ADDW6 = 000000           722     773
ADDW7 = 000000           722     774
ADDW8 = 000000           722     775
ADDW9 = 000000           722     776
ADEVCT= 000000           722     728
ADEVM = 000000           722     764
ADR     017024           867#    1110    1125    1146*   1147*   1152    1155*   1156*   1194*   1229*   1232*   1234*   1930*
                        1933    1939    1940    1951    2307*   2310*   2312*   2465*   2468*   2470*   2622*   2625*   2627*
AENV  = 000000           722     733
AENVM = 000000           722     734
AFATAL= 000000           722     725
AMADR1= 000000           722     750
AMADR2= 000000           722     754
AMADR3= 000000           722     757
AMADR4= 000000           722     760
AMAMS1= 000000           722     744
AMAMS2= 000000           722     752
AMAMS3= 000000           722     755
AMAMS4= 000000           722     758
AMSGAD= 000000           722     730
AMSGLG= 000000           722     731
AMSGTY= 000000           722     724
AMTYP1= 000000           722     745
AMTYP2= 000000           722     753
AMTYP3= 000000           722     756
AMTYP4= 000000           722     759
APASS = 000000           722     727
APRIOR= 000000           722
APTCSU= 000040          3505#
APTENV= 000001          3382    3461    3503#
APTSIZ= 000200          1059    3502#
APTSPO= 000100          3463    3504#
ASWREG= 000000           722     735
ATESTN= 000000           722     726
AUNIT = 000000           722     729
AUSWR   000000           722     736
```

```
AVECT1= 000000            722     761
AVECT2= 000000            722     762
BAD     025124           1911    1919#   1928
BEGIN   021420           1221    1237#
BFADR   017034            871#   2112    2117    2121    2127    2146    2151    2158    2163
BIT0  = 000001            196#    500    1073    1132    1134    1147    1156    1230    2308    2466    2623    2869
BIT00 = 000001            186#    196
BIT01 = 000002            185#    195
BIT02 = 000004            184#    194
BIT03 = 000010            183#    193
BIT04 = 000020            182#    192    2853
BIT05 = 000040            181#    191
BIT06 = 000100            180#    190
BIT07 = 000200            179#    189
BIT08 = 000400            178#    188
BIT09 = 001000            177#    187    3318
BIT1  = 000002            195#    501
BIT10 = 002000            176#    538
BIT11 = 004000            175#    537    1110    1152
BIT12 = 010000            174#    536
BIT13 = 020000            173#
BIT14 = 040000            172#    511    1803    3302
BIT15 = 100000            171#    512    1108    1192    1210    1214    1218    1236    1778    1868    1971    2040    2197
                         2248    2355    2406    2512    2563    2816
BIT2  = 000004            194#    502    1194    1930
BIT3  = 000010            193#    503
BIT4  = 000020            192#    544    1290    1334    1363    1398    1434    1440    1460    1468    1499    1506    1530
                         1537    1548    1574    1598    1606    1628    1636    1658    1666    1688    1696    1722    1728
                         2909
BIT5  = 000040            191#    509     543    1799
BIT6  = 000100            190#    508     542    2889
BIT7  = 000200            189#    541    2861    2867    2880    2892
BIT8  = 000400            188#    540    1810    3723
BIT9  = 001000            187#    510     539    1166    1184    1806    1829    1836    2804    3374
BMSK    017032            870#   1032*   1146    1163*
BOOT    014526            795#   2242*   2266    2287    2400*   2424    2445    2557*   2581    2602    2747
BOXNUM  014646            822#   2264    2422    2579    2692*   2708*   2720
BPTVEC= 000014            203#
BUFCLR  032322           2279    2437    2594    2729#   2732
CACHVE= 000114            211#   1268*
CKSUM   014650            823#   1324*   1325*   1326*   1345    2115*   2118*   2119*   2131    2149*   2152*   2153*   2167
CONTRL= 177746            220#   1166*   1174*   1184*   2804*   2920    2980*   2995*   2996*   3060    3119*
COUNT0  016754            849#   2936*   2945*   2947    3007    3076*   3085*   3087
COUNT1  016756            850#   2950*   2951    3090*   3091
COUNT2  016760            851#   2954*   2955    3094*   3095
COUNT3  016762            852#   2958*   2959    3098*   3099
CPUACT  016766            854#   1023    1107*   1157*   1230    1772    1846    1871    1874    1974    1977    2043    2046
                         2069    2079    2189    2308    2347    2466    2504    2623    2684    3175    3267    3271*
CPUER   040362           1066    1072    1078    1443    1473    1510    1541    1610    1640    1670    1700    1734    1800
                         2201    2252    2359    2410    2516    2567    2719    2754    2776    3813#
CPUERR= 177766            232#   2832    3814
CP0   = 000001            500#
CP1   = 000002            501#
CP2   = 000004            502#
CP3   = 000010            503#
CR    = 000015            108#   3726    3819    3828    3832    3838    3845    3852    3863    3870    3879    3891    3905
```

```
                    3911    3915    3925    3929    3936    3941    3947    3995    4240    4246    4253    4260    4265
                    4269    4274    4279    4284    4287    4292    4298    4302    4308    4316    4323    4327    4333
                    4339    4356    4361    4368    4372    4387    4392    4400    4407    4429
CRLF   = 000200      109#
C1       016722     837#    1892
C2       016724     838#    1893
C3       016776     858#    1894
DCF    = 000007     488#    1124    1932    1944
DDISP  = 177570     102#     673     674     675     676
DH10     046346     896      938     950     968     974    1010    1016    4368#
DH11     046371     944     4372#
DH12     046423     956     4377#
DH14     046512     962      980    4387#
DH20     046545     986     4392#
DH21     046622     992     4400#
DH22     046672     998    1004    4407#
DH5      046244     920     926     4356#
DH7      046277     932     4361#
DIGITS   037372    3542*   3543*   3544*   3545    3546*   3560    3562*   3577*   3590#
DISPLA   014170     673#    1068*   1266*   1320*   1355*   1390*   1425*   1450*   1490*   1519*   1565*   1589*   1619*
                    1649*   1679*   1709*   1737*   1863*   1967*   2036*   2182*   2198*   2249*   2340*   2356*   2407*
                    2497*   2513*   2564*   2655*   3333*   3352*
DISPRE   000174     552#    1068
DSKBUF   051052     871     4433#
DSWR   = 177570     101#     669     670     671     672
DT10     046752     897      939     951     969     975    1011    1017    4418#
DT11     046760     945      963     981    4419#
DT12     046770     957     4421#
DT20     047000     987     4423#
DT21     047014     993     4425#
DT22     047026     999     1005    4427#
DT5      046730     921      927    4414#
DT7      046740     933     4416#
D1       016726     839#    1995
D2       016730     840#    1996
D3       017000     859#    1997
EMTVEC=  000030     206#    1043*   1044*
EM1      045057     895     4240#
EM10     045400     937     4279#
EM11     045434     943     4284#
EM12     045456     949     4287#
EM13     045510     955     4292#
EM14     045551     961     4298#
EM15     045577     967     4302#
EM16     045636     973     4308#
EM17     045715     979     4316#
EM2      045121     901     4246#
EM20     045764     985     4323#
EM21     046010     991     4327#
EM22     046051     997     4333#
EM23     046112    1003     4339#
EM24     046162    1009     4346#
EM25     046211    1015     4350#
EM3      045167     907     4253#
EM4      045234     913     4260#
EM5      045270     919     4265#
```

```
EM6     045317              925    4269#
EM7     045347              931    4274#
END     111052             3571    4434#
ENTR    032762             2293    2451    2608    2804#
ENTR22  014542              801#   2187*   2189
ENTR23  014544              802#   2345*   2347
ENTR24  014546              803#   2502*   2504
EOPLID  035760             3205    3214    3223    3232    3281#
ERRBUF  047052             3569    3573    4432#
ERRDIS  040356             1065    1071    1077    3811#
ERRLCK  016774              857#   3396*   3440*   3667
ERRTAB  014702              830#   1066*   1072*   1078*   1427*   1443*   1452*   1473*   1492*   1510*   1522*   1541*   1591*
                           1610*   1621*   1640*   1651*   1670*   1681*   1700*   1712*   1734*   1797*   1800*   2201*   2252*
                           2359*   2410*   2516*   2567*   2694*   2719*   2754*   2776*   3811
ERRVEC= 000004              199#   1049*   1065*   1071*   1077*   3307    3308*   3310*   3313*
EXC   = 000010              489#
EXIT    014540              800#   2191*   2203    2244    2257*   2349*   2361    2402    2415*   2506*   2518    2559    2572*
                           2725*
EXTRA   016764              853#   1029
E1      016732              841#   2064    2069
E2      016734              842#   2065
E3      017002              860#   2066
FACTOR  017060              877#
FIRST   017030              869#   1031*
FLAG    014602              812#   1521*   1542    3808*   3809*
FLAGB   016736              843#   3268    3270*
FORKTB  021222             1199    1204#
GNS   = ****** U            551    3188    3195    3202    3211    3220    3229    3238    3435    3798    3799    3800    3801
                           3802
HERE    040052             3714#   3747
HIADRS= 177742              218#
HIBOX   014550              804#   2210    2211    2212    2272    2273    2274    2368    2369    2370    2430    2431    2432
                           2525    2526    2527    2587    2588    2589   2711*   2777    2779    2782    3026    3027    3028
HICORE  014532              797#   2270*   2328*   2428*   2486*   2585*   2643*   3014
HITMIS= 177752              222#
HOE   = 100000              512#
HT    = 000011              106#
ILLDN   035054             3103    3141#
ILLDWN  034176             2964    3002#
ILLUP   034172             2906    2999#
IMSK  = 000004              485#
INTMSK  016772              856#   1155    1164*
IOTVEC= 000020              204#
ISTDIS  040334             1848    3806#
ISTTAB  014672              828#   1911*   1928*   1934*   3806
ISTVEC  017026              868#   1127*   1222    1224*   1848*   2293*   2451*   2608*
KDPAR0= 172360              374#
KDPAR1= 172362              375#
KDPAR2= 172364              376#
KDPAR3= 172366              377#
KDPAR4= 172370              378#
KDPAR5= 172372              379#
KDPAR6= 172374              380#
KDPAR7= 172376              381#   2924    2975    3064    3114
KDPDR0- 172320              352#
KDPDR1= 172322              353#
```

```
KDPDR2= 172324   354#
KDPDR3= 172326   355#
KDPDR4= 172330   356#
KDPDR5= 172332   357#
KDPDR6= 172334   358#
KDPDR7= 172336   359#
KERSTK= 013776    92#
KIPAR0= 172340   363#  2210*  2272*  2368*  2430*  2525*  2587*  2735*  2841*  3016*  3026*
KIPAR1= 172342   364#  2211*  2273*  2369*  2431*  2526*  2588*  2737*  2843*  3018*  3027*
KIPAR2= 172344   365#  2212*  2274*  2370*  2432*  2527*  2589*  2739*  2845*  3020*  3028*
KIPAR3= 172346   366#  2752*  2757    2764    2769*  2777*  2794*
KIPAR4= 172350   367#  2779*  2780*  2795*
KIPAR5= 172352   368#  2782*  2783*  2796*  2847*
KIPAR6= 172354   369#  2849*
KIPAR7= 172356   370#  2741*  2851*  3022*
KIPDR0= 172300   341#  2736*  2842*  3017*
KIPDR1= 172302   342#  2738*  2844*  3019*
KIPDR2= 172304   343#  2740*  2846*  3021*
KIPDR3= 172306   344#  2753*  2778*
KIPDR4= 172310   345#  2781*
KIPDR5= 172312   346#  2784*  2848*
KIPDR6= 172314   347#  2850*
KIPDR7= 172316   348#  2742*  2852*  3023*
LF    = 000012   107#  3728    3819    3828    3832    3838    3845    3852    3863    3870    3879    3891    3905
                3911    3915    3925    3929    3936    3941    3947    3995    4240    4246    4253    4260    4265
                4269    4274    4279    4284    4287    4292    4298    4302    4308    4316    4323    4327    4333
                4339    4356    4361    4368    4372    4387    4392    4400    4407    4429
LKS   = 177546   103#  2889*  2892*  2896*
LKVEC = 000100   210#
LOADRS= 177740   217#
LOE   = 001000   510#
LOOPS   016752   848#  2943*  2947    2951    2955    2959    3007    3083*  3087    3091    3095    3099
LOT   = 040000   511#
MAINT = 177750   221#
MAP     033142  1713    2841#
MAPH0 = 170202   457#
MAPH00= 170202   393#   457
MAPH01= 170206   395#   459
MAPH02= 170212   397#   461
MAPH03= 170216   399#   463
MAPH04= 170222   401#   465
MAPH05= 170226   403#   467
MAPH06= 170232   405#   469
MAPH07= 170236   407#   471
MAPH1 = 170206   459#
MAPH10= 170242   409#
MAPH11= 170246   411#
MAPH12= 170252   413#
MAPH13= 170256   415#
MAPH14= 170262   417#
MAPH15= 170266   419#
MAPH16= 170272   421#
MAPH17= 170276   423#
MAPH2 = 170212   461#
MAPH20= 170302   425#
MAPH21= 170306   427#
```

```
MAPH22= 170312          429#
MAPH23= 170316          431#
MAPH24= 170320          433#
MAPH25= 170326          435#
MAPH26= 170332          437#
MAPH27= 170336          439#
MAPH3 = 170216          463#
MAPH30= 170342          441#
MAPH31= 170346          443#
MAPH32= 170352          445#
MAPH33= 170356          447#
MAPH34= 170362          449#
MAPH35= 170366          451#
MAPH36= 170372          453#
MAPH37= 170376          455#    2928    2971    3068    3110
MAPH4 = 170222          465#
MAPH5 = 170226          467#
MAPH6 = 170232          469#
MAPH7 = 170236          471#
MAPL0 = 170200          456#
MAPL00= 170200          392#    456     2926    2973    3066    3112
MAPL01= 170204          394#    458
MAPL02= 170210          396#    460
MAPL03= 170214          398#    462
MAPL04= 170220          400#    464
MAPL05= 170224          402#    466
MAPL06= 170230          404#    468
MAPL07= 170234          406#    470
MAPL1 = 170204          458#
MAPL10= 170240          408#
MAPL11= 170244          410#
MAPL12= 170250          412#
MAPL13= 170254          414#
MAPL14= 170260          416#
MAPL15= 170264          418#
MAPL16= 170270          420#
MAPL17= 170274          422#
MAPL2 = 170210          460#
MAPL20= 170300          424#
MAPL21= 170304          426#
MAPL22= 170310          428#
MAPL23= 170314          430#
MAPL24= 170320          432#
MAPL25= 170324          434#
MAPL26= 170330          436#
MAPL27= 170334          438#
MAPL3 = 170214          462#
MAPL30= 170340          440#
MAPL31= 170344          442#
MAPL32= 170350          444#
MAPL33= 170354          446#
MAPL34= 170360          448#
MAPL35= 170364          450#
MAPL36= 170370          452#
MAPL37= 170374          454#
MAPL4 = 170220          464#
```

```
MAPL5 = 170224   466#
MAPL6 = 170230   468#
MAPL7 = 170234   470#
MBDSW   014632   818#  1812*  1815*  1825*  1835*  1836*  2084   2107   2139   2863
MBUSR   033274  2114   2123   2148   2160   2860#
MEMERR= 177744   219#  2833   2836*
MEMSIZ  032076  2243   2401   2558   2690#
MMERR   033270  2744   2854   2856#  2857   2870*  2871*  2872*  2873*  2874*  2875*  2876*  2877*  2878*
                2879*  3025
MMR0    177572   242#   246   1716*  1732*  2205*  2213*  2218*  2275*  2323   2327*  2363*  2371*  2376*
                2433*  2481   2485*  2520*  2528*  2533*  2590*  2638   2642*  2755*  2772*  2787*  2791*
                2915   2984   3029*  3055   3123
MMR1  = 177574   243#   247
MMR2  = 177576   244#   248
MMR3  = 172516   245#   249   2743*  2853*  2921   2979*  3024*  3061   3118*
MMVEC = 000250   213#  1714*  2744*  2854*  3025*
MPI     016737   844#  1073*  1079*  1102   1220   1738   2993   3162   3197   3675
MPSW  = 000100   508#  1062   1069
MSTENT  021174  1182   1196#
MSO     021312  1204   1220#
MTC   = 000015   490#
NCX   = 010000   536#  1910   2012   2103
NODEV   043732  2088   4134#
NOPRMP  017004   861#  1957*  2226*  2319*  2384*  2477*  2541*  2634*  2821*  3399*  3682   3685*
NORH70  024466  1822#  1852
NORP    024634  1797   1851#
NS    = 002000   538#
NXTDRV  024450  1804   1809   1811   1817#
OK      047036  2302   2460   2617   4429#
PARERR  033074  1268   2828#
PARFLG  033076  1269*  2829#  2837*
PATCHK  014530   796#  2277*  2321*  2435*  2479*  2592*  2636*  2759
PATTRN  032434  2278   2322   2436   2480   2593   2637   2747#
PFDT    014622   816#
PFFT    014612   814#  1711*  1910*  2012*  2103*  2200*  2251*  2358*  2409*  2515*  2566*  2676*  3046
                3127
PGCS  = 000001   482#  1151   1193   1929
PGF   = 000005   486#
PGTE  = 000000   481#  1145   1226   1938   2306   2464   2621
PIRQ  = 177772   100#
PIRQVE= 000240   212#
POWDWN  033530  1267   1471   1551   2906#  2988
POWUP   034012  2942   2963#
PR0   = 000000   133#
PR1   = 000040   134#
PR2   = 000100   135#
PR3   = 000140   136#
PR4   = 000200   137#
PR5   = 000240   138#
PR6   = 000300   139#
PR7   = 000340   140#
PS    = 177776    97#    98   1283*  1293*  1321*  1356*  1366*  1391*  1401*  1426*  1451*  1462*  1491*
               1502*  1520*  1532*  1566*  1590*  1601*  1603*  1620*  1631*  1633*  1650*  1661*  1663*
               1680*  1691*  1693*  1710*  2891*
PSW   = 177776    98#  1195*  2192*  2350*  2507*  2919   2981*  3059   3120*
PUT     017054   875#  1033*  1897*  1945   1999*  2068*
```

```
PWRDIS  040340          1047    1270    1479    3808#
PWRFL   014522           793#   2222    2241*   2298    2315    2380    2399*   2456    2473    2537    2556*   2613    2630
                        2722
PWRTAB  014662           826#   1267*   1464*   1471*   1544*   1551*   1796*   2818*   2906*   2942*   2964*   2988*   3042
                        3045*   3077*   3103*   3131*   3810
PWRVEC= 000024           205#   1047*   1048*   1270*   1271*
P21     026216          2096    2099#
RELODN  014536           799#   2216    2219*   2326*   2374    2377*   2484*   2531    2534*   2641*
RELOHI  032576          2268    2426    2583    2775#
RELOLO  032722          2325    2483    2640    2794#
RELOUP  014534           798#   2207    2214*   2271*   2365    2372*   2429*   2522    2529*   2586*
RESTAB  036616          3171    3278    3443#
RESTRT  020000           568    1022#   3269
RESVEC= 000010           200#
ROUTE   014562           807#   3778*   3780
RPAS  = 176716           521#
RPBA  = 176704           516#   2865*
RPCC  = 176736           529#
RPCS1 = 176700           514#   2113*   2122*   2147*   2159*   2861    2869*   2882
RPCS2 = 176710           518#   1799*   1802*   1813    1828*   2863*
RPDA  = 176706           517#   2866*
RPDB  = 176722           523#
RPDC  = 176734           528#
RPDS  = 176712           519#   1803    1806    1810    1829    2867    2880
RPDT  = 176726           525#
RPEC1 = 176744           532#
RPEC2 = 176746           533#
RPER1 = 176714           520#
RPER2 = 176740           530#
RPER3 = 176742           531#
RPLA  = 176720           522#
RPMR  = 176724           524#
RPOF  = 176732           527#
RPPGM   016742           846#   1808*   1820
RPSN  = 176730           526#
RPSRC   024364          1802#   1819
RPWC  = 176702           515#   2864*
SAVRG   014552           805#   3772*   3779
SAV6    014572           810#   2935*   2965    3075*   3104
SDPAR0= 172260           330#
SDPAR1= 172262           331#
SDPAR2= 172264           332#
SDPAR3= 172266           333#
SDPAR4= 172270           334#
SDPAR5= 172272           335#
SDPAR6= 172274           336#
SDPAR7= 172276           337#
SDPDR0= 172220           308#
SDPDR1= 172222           309#
SDPDR2= 172224           310#
SDPDR3= 172226           311#
SDPDR4= 172230           312#
SDPDR5= 172232           313#
SDPDR6= 172234           314#
SDPDR7= 172236           315#
SEC2    024170          1274    1767#   1769
```

```
SED    = 000200            541#
SETCLK  033440            2887#
SETMM   032340            2209    2367    2524    2735#   2751    2775
SEU    = 000100            542#
SID    = 001000            539#
SIGNAL  014642             820#   1913*   1935*   2014*   2029*   2105*   2134*   2'70*   3048*   3129*
SIPAR0= 172240             319#
SIPAR1= 172242             320#
SIPAR2= 172244             321#
SIPAR3= 172246             322#
SIPAR4= 172250             323#
SIPAR5= 172252             324#
SIPAR6= 172254             325#
SIPAR7= 172256             326#
SIPDR0= 172200             297#   2922    2977    3062    3116
SIPDR1= 172202             298#
SIPDR2= 172204             299#
SIPDR3= 172206             300#
SIPDR4= 172210             301#
SIPDR5= 172212             302#
SIPDR6= 172214             303#
SIPDR7= 172216             304#
SIU    = 000400            540#
SIZEHI= 177762             229#
SIZELO= 177760             227#
SIZEND  024600            1816    1826    1844#
SIZMBS  024340            1797#
SLVENT  021110            1127    1183#
SLVID   014644             821#   1185*   1186
SL1     021232            1205    1208#
SL2     021252            1206    1212#
SL3     021272            1207    1216#
SR0    = 177572            246#
SR1    = 177574            247#
SR2    = 177576            248#
SR3    = 172516            249#
SSD    = 000040            543#   3046
SSU    - 000020            544#   1910    2012    2103    3127
STACK  = 013776             91#     92      93      94     677     678     679     680
START   014502             790#   2259    2417    2574    2699*   2700*   2701    2703*   2707    2709    2711    2713*   2749
                          2752
STCS   - 000003            484#
STF    = 000006            487#
STKLMT= 177774              99#
STOP    014462             787#   1027    2696*   2697*   2698*   2704    2706*   2707*   2757
STRT    020064             566     569    1037#
STS17   025264            1934    1943#
STTE   = 000002            483#
STO     021046            1104    1174#
SUPSTK= 013576              93#
SWR     014160             669#   1058*   1061*   1062    1067*   1069    1083    1084    1089    1093    1095    1106    1177
                          1227    1272    1312    1349    1382    1416    1444    1482    1511    1557    1581    1611    1641
                          1671    1701    1735    2304    2462    2619    2657    3302    3318    3359    3374
SWREG   000176             553#
SW0     000001             168#
SW00   - 000001             158#    168
```

```
SW01   = 000002          157#    167
SW02   = 000004          156#    166
SW03   = 000010          155#    165
SW04   = 000020          154#    164
SW05   = 000040          153#    163
SW06   = 000100          152#    162
SW07   = 000200          151#    161    1095    1272
SW08   = 000400          150#    160    2657
SW09   = 001000          149#    159
SW1    = 000002          167#
SW10   = 002000          148#
SW11   = 004000          147#
SW12   = 010000          146#
SW13   = 020000          145#
SW14   = 040000          144#   1312    1349    1382    1416    1444    1482    1511    1557    1581    1611    1641    1671
                        1701    1735
SW15   = 100000          143#
SW2    = 000004          166#
SW3    = 000010          165#
SW4    = 000020          164#
SW5    = 000040          163#
SW6    = 000100          162#
SW7    = 000200          161#
SW8    = 000400          160#
SW9    = 001000          159#
SYNC.1   014712          832#   1914*   1925*   1959*   2015*   2020*   2104*   2155*   3130*
SYNC.2   014714          833#   1884*   1987*   2056*   2093*   2110*   2136*   2141*   2678*   2682*   2684
SYNC.3   016770          855#   1768*   1771*   2185*   2188*   2343*   2346*   2500*   2503*   2680*   2683*   3160*   3167*
                        3174*
SYSTID=  177764          231#
S2LOG1   016716          835#   1770*   1772
S2LOG2   016720          836#   1845*   1846
S2L1     017020          865#   1025    1780*   1844*
TBITVE=  000014          201#
TI     = 004000          537#   1711    1910    2012    2103    2676
TIMIT    034202         2987    3006#   3126
TKVEC  = 000060          208#
TM1      040376         1056    3819#
TM10     040726         3858#
TM100    041507         1906    2008    2099    3925#
TM101    041533         1822    3929#
TM102    041604         2092    3936#
TM103    041635         2227    2320    2385    2478    2542    2635    3941#
TM104    041700         2675    3947#
TM106    042317         2280    2438    2595    3995#
TM107    042363         2443    4002#
TM108    043011         2285    4050#
TM109    043771         2724    4140#
TM11     040761         1138    3863#
TM110    044053         2600    4149#
TM111    044746         1958    2822    4227#
TM12     040772         1141    3865#
TM13     041024         1280    3870#
TM14     041370         1277    3911#
TM15     041420         1776    3915#
TM2      040461         1088    3828#
```

```
TM4     040506            1097    3832#
TM5     040550            1100    3838#
TM6     040616            1082    3845#
TM7     040663            1080    3852#
TM76    041450            1116    3919#
TM77    041476            1900    2002    2073    2235    2393    2550    2668    3923#
TPVEC = 000064             209#
TQL1    017016             864#   3686*   3702*   3712*   3752*
TRAPVE= 000034             207#   1045*   1046*
TRTVEC= 000014             202#
TST1    021420            1200    1201    1202    1203    1211    1215    1219    1265#   1313    3443    3444    3445    3446
TST10   022732            1518#   1558
TST11   023130            1564#   1582
TST12   023222            1588#   1612
TST13   023356            1618#   1642
TST14   023512            1648#   1672
TST15   023646            1678#   1702
TST16   024002            1708#   1736
TST17   024644            1849    1862#   1916    1921    1942    1961
TST2    021654            1319#   1350
TST20   025362            1880    1966#   2017    2031
TST21   025704            1983    2035#   2094    2111    2137    2144    2173
TST22   026632            2052    2181#
TST23   027654            2206    2246    2258    2339#
TST24   030676            2364    2404    2416    2496#
TST25   031720            2521    2561    2573    2654#
TST3    022014            1354#   1383
TST4    022146            1389#   1417
TST5    022300            1424#   1445
TST6    022414            1449#   1483
TST7    022606            1489#   1512
TS17A   025014            1888    1896#
TS17B   025136            1890    1924#   1926
TS20A   025532            1991    1998#
TS20B   025634            1993    2019#   2021
TS21A   026054            2060    2067#
TS21B   026434            2062    2138#
TS22A   027106            2195    2232#
TS22B   026716            2194    2197#   2215    2217    2220    2225    2228
TS23A   030130            2353    2390#
TS23B   027740            2352    2355#   2373    2375    2378    2383    2386
TS24A   031152            2510    2547#
TS24B   030762            2509    2512#   2530    2532    2535    2540    2543    2823
TS25A   031766            2663    2665#
TS25B   032020            2664    2676#
TYPDS = 104405            1118    1140    1824    1839    1842    1908    2010    2090    2101    2283    2441    2598    3192
                          3207    3216    3225    3234    3242    3802#
TYPE  = 104401             1#     1056    1080    1082    1088    1091    1092    1097    1100    1116    1137    1138    1141
                          1277    1280    1281    1776    1822    1837    1840    1843    1900    1906    1909    1927    1958
                          2002    2008    2011    2022    2073    2088    2091    2092    2099    2102    2143    2157    2172
                          2227    2235    2280    2285    2302    2320    2385    2393    2438    2443    2460    2478    2542
                          2550    2595    2600    2617    2635    2668    2674    2675    2686    2724    2822    3009    3186
                          3193    3199    3200    3208    3209    3217    3218    3226    3227    3235    3236    3243    3363
                          3405    3407    3412    3414    3433    3438    3568    3650    3798#
TYPIT   040236            3727    3729    3731    3733    3735    3748    3760#   3761
TYPLCK  017056             876#   1034*   3398*   3439*   3669
```

```
TYPOC = 104402            1090    3431    3799#
TYPON = 104404            3801#
TYPOS = 104403            1903    2005    2076    2238    2396    2553    2671    3800#
TYPQUE  014716             834#   1050*   1051*   3264    3265    3273*   3274*   3689    3692*   3696    3699*   3716    3718
                          3721*   3738    3740    3743*   3750*   3751*
UBCR2 = 170016             472#   1290*   1334*   1363*   1398*   1434*   1440*   1460*   1468*   1499*   1506*   1530*   1537*
                          1548*   1574*   1598*   1606*   1628*   1636*   1658*   1666*   1688*   1696*   1722*   1728*   2909*
UBEF    016740             845#   1098*   1101*   1278    1285    1303    1329    1358    1372    1393    1407    1429    1438
                          1455    1466    1494    1504    1525    1535    1546    1569    1593    1604    1623    1634    1653
                          1664    1683    1694    1717    1726    2907    2989    3266    3272*
UBELCK  017014             863#   1287*   1331*   1360*   1395*   1431*   1457*   1496*   1527*   1571*   1595*   1625*   1655*
                          1685*   1719*   2991*
UBESW = 000040             509#   1093
UDPAR0= 177660             286#
UDPAR1= 177662             287#
UDPAR2= 177664             288#
UDPAR3= 177666             289#
UDPAR4= 177670             290#
UDPAR5= 177672             291#
UDPAR6= 177674             292#
UDPAR7= 177676             293#   2917    2982    3057    3121
UDPDR0= 177620             264#
UDPDR1= 177622             265#
UDPDR2= 177624             266#
UDPDR3= 177626             267#
UDPDR4= 177630             268#
UDPDR5= 177632             269#
UDPDR6= 177634             270#
UDPDR7= 177636             271#
UIPAR0= 177640             275#
UIPAR1= 177642             276#
UIPAR2= 177644             277#
UIPAR3= 177646             278#
UIPAR4= 177650             279#
UIPAR5= 177652             280#
UIPAR6= 177654             281#
UIPAR7= 177656             282#
UIPDR0= 177600             253#
UIPDR1= 177602             254#
UIPDR2= 177604             255#
UIPDR3= 177606             256#
UIPDR4= 177610             257#
UIPDR5= 177612             258#
UIPDR6= 177614             259#
UIPDR7= 177616             260#
UNIEOP  035564            3198    3240#
USESTK= 013476              94#
WAIT  = ****** U          3177
YELLIM  017044             873#   3051
YYY     014524             794#   3422*   3425*   3426    3429*   3430
$AFRK = 000000               1#
$APTHD  000230             580     586#
$ASTAT= ****** U          3483    3498
$ATYC   036652            3454    3456#
$ATY1   036626            3452#
$ATY3   036634            3453#
```

```
$ATY4   036644            3385    3455#
$AUTOB  014154             666#
$BASE   014412             763#
$BDADR  014114             647#
$BDDAT  014140             660#
$CDW1   014416             765#
$CDW2   014420             766#
$CKSWR= ****** U           3805
$CMTAG  014000             600#    1175
$CM1  = 000005             696#     698#    700#    702#    704#    706#
$CM2  = 000012             696#     698#    700#    702#    704#    706#
$CM3  = 000005             694#     696
$CM4  = 000005             706#     707#    708#    709#    710#    711#
$CPUID  014226             690#    1038    1119*    1142    1191*    1788    1823    1841    1907    1946    2009    2089    2100
                          2808    3034    3283
$CPUOP  014364             737#
$CRLF   014333             716#    1091    1092    1137    1281    1909    2011    2091    2102    2674    3199    3208    3217
                          3226    3235    3243    3407    3414    3438
$DBLK   037614            3615    3650    3659#
$DDW0   014422             767#
$DDW1   014424             768#
$DDW10  014446             777#
$DDW11  014450             778#
$DDW12  014452             779#
$DDW13  014454             780#
$DDW14  014456             781#
$DDW15  014460             782#
$DDW2   014426             769#
$DDW3   014430             770#
$DDW4   014432             771#
$DDW5   014434             772#
$DDW6   014436             773#
$DDW7   014440             774#
$DDW8   014442             775#
$DDW9   014444             776#
$DEVCT  014346             728#
$DEVM   014414             764#
$DOAGN  035654            3182    3254    3257    3263#
$DOWN   045004            3009    4232#
$DTBL   037604            3618    3655#
$ENDAD  035644             560    3259#   3390
$ENDCT  035200            1180    3184#
$ENULL  035754            3279#
$ENV    014356             733#    3382    3461    3485
$ENVM   014357             734#    1059    3463    3662
$EOP    035060            1740    2659    2687    3159#   3161
$EOPCT  035166            1180*   3180#   3185
$EOPSG  014134             655#   3166*   3168    3173*   3175    3276*
$ERFLG  014006             606#    3295    3316    3323*   3336    3350*
$ERGBL  014224             689#    3241   3244*   3354*
$ERMAX  014060             631#   3332*   3336
$ERROR  036164            1043    3349#
$ERRPC  014064             635#   3355*   3356*   3357    4414    4416    4418    4419    4421    4427
$ERRSP  014074             639#   3358*
$ERRTB  017062             893#    3403
$ERRTY  036404            3394#
```

```
$ERTTL  014042            622#  3206  3215  3224  3233  3245  3353*
$ESCAP  014322            711#  1197* 3331* 3377  3379
$ETABL  014356            732#
$ETEND  014462            592   785#
$FATAL  014340            725#  3489*
$FFLG   037072           3452*  3455* 3483  3492* 3500#
$FILLC  014222            687#
$FILLS  014221            686#
$GDADR  014104            643#
$GDDAT  014124            651#
$GET    035634           3252   3256#
$GET42  035624           3253#
$GTSWR= ****** U         3804
$HD   = 000000             11
$HIBTS  000230            587#
$ICNT   014012            610#
$ILLUP  035050           3045   3137#
$INTAG  014155            667#
$ITEMB  014054            627#  3357* 3384  3400
$LF     014334            717#
$LFLG   037071           3493*  3499#
$LPADR  014022            614#  3321* 3329* 3334  3336
$LPERR  014032            618#  1198* 3321  3330* 3336  3376
$MADR1  014370            750#
$MADR2  014374            754#
$MADR3  014400            757#
$MADR4  014404            760#
$MAIL   014336            588    592   723#  3325
$MAMS1  014366            744#
$MAMS2  014372            752#
$MAMS3  014376            755#
$MAMS4  014402            758#
$MBADR  000232            588#
$MFLG   037070           3453*  3459  3494* 3498#
$MSGAD  014352            730#  3469* 3472
$MSGLG  014354            731#  3474*
$MSGTY  014336            724#  3467  3475* 3487  3491*
$MTYP1  014367            745#
$MTYP2  014373            753#
$MTYP3  014377            756#
$MTYP4  014403            759#
$NULL   014220            685#  1927  2022  2143  2157  2172  2686
$NWTST= 000001           1262#  1316# 1351# 1386# 1421# 1446# 1486# 1515# 1561# 1585# 1615# 1645# 1675#
                         1705#  1859# 1963# 2032# 2178# 2336# 2493# 2651#
$OCNT   037366           3538*  3574* 3587#
$OMODE  037370           3533*  3537* 3542  3545* 3556* 3589#
$OVER   036150           3303   3322  3333#
$PASS   014344            727#  1181* 3177* 3178* 3190  3279
$PASTM  000236            590#
$PGM1   043535           1837   4111#
$PGM2   043563           1840   4115#
$PGM3   043577           1843   4118#
$POWER  034220           1782   3013#
$PSWR   014660            825#  1083* 1086  3263  3275*
$PWRDN  034412           1796   2818  3045# 3131
$PWRUP  034676           3077   3103#
```

```
$QUES   014332            715#
$RDCHR= ****** U         3805
$RDDEC= ****** U         3805
$RDLIN= ****** U         3805
$RDOCT= ****** U         3805
$REGAD  014236            694#
$REG0   014240            696#  1786*  4414   4416   4419   4423   4425   4427
$REG1   014250            698#  1937*  1954*  2764*  2832*  3814*  4414   4416   4421   4423   4425
$REG2   014260            700#  1939*  1955*  2833*  4416   4423   4425
$REG3   014270            702#  1940*  4423
$REG4   014300            704#
$R2A  = ****** U         3805
$SAVRE= ****** U         3805
$SCOPE  036004           3301#
$SETUP= 000034            571#  3164   3302
$STUP = 177777            571#
$SVLAD  036110           3311   3317   3324#
$SVPC = 000200            558#   563
$SWR  = 141000              1#    11    711    715   1267   1321   1356   1391   1426   1451   1491   1520   1566
                         1590   1620   1650   1680   1710   1864   1968   2037   2183   2341   2498   2656   3156
                         3164   3255   3278   3279   3295   3296   3297   3298   3302   3314   3316   3317   3318
                         3324   3330   3333   3336
$SWREG  014360            735#  1061   1067
$SWRMK= 000000           3298
$TESTN  014342            726#  3327*
$TKB    014212            682#  2731
$TKS    014210            681#  2291   2296   2449   2454   2606   2611   2729
$TMP0   014310            706#  1136*  1139
$TMP1   014312            707#
$TMP2   014314            708#
$TMP3   014316            709#
$TMP4   014320            710#
$TN   - 000026              1#    11   1262   1266   1267#  1312   1316   1320   1321#  1349   1351   1355   1356#
                         1382   1386   1390   1391#  1416   1421   1425   1426#  1444   1446   1450   1451#  1482
                         1486   1490   1491#  1511   1515   1519   1520#  1557   1561   1565   1566#  1581   1585
                         1589   1590#  1611   1615   1619   1620#  1641   1645   1649   1650#  1671   1675   1679
                         1680#  1701   1705   1709   1710#  1735   1859   1863   1864#  1963   1967   1968#  2032
                         2036   2037#  2178   2182   2183#  2336   2340   2341#  2493   2497   2498#  2651   2655
                         2656#
$TPB    014216            684#  3762*
$TPFLG  014223            688#
$TPS    014214            683#  3760
$TRAP   040252           1045   3772#
$TRAP2  040306           3786#  3797
$TRP  = 000006           3790#  3799#  3800#  3801#  3802#  3803#
$TRPAD  040320           3777   3797#
$TSTM   000234            589#
$TSTNM  014002            602#  1896*  1902   1924*  1998*  2004   2019*  2067*  2075   2138*  2184*  2237   2342*
                         2395   2499*  2552   2656*  2670   3295   3324*  3327   3333   3336   3352   3423   4418
                         4419   4421   4423   4425   4427
$TYPBN= ****** U         3803
$TYPDS  037374           3603#  3802
$TYPE   037624           3480   3661#  3670   3790   3798
$TYPOC  037120           3536#  3799
$TYPON  037134           3535   3538#  3801
$TYPOS  037074           3531#  3800
```

```
$UNIT   014350            729#
$UNITM  000240            591#
$USWR   014362            736#
$VECT1  014406            761#
$VECT2  014410            762#
$XTSTR  036014           3305#
$$GET4= 000001           3255#  3256#
$$STP   014200            677#  1037   1179   1208   1212   1216   1296   1301   1348   1368   1369   1403   1404
                         1436   1472   1501   1509   1568   1580   1600   1609   1630   1639   1660   1669   1690
                         1699   1724   1733   1779   1867   1970   2039   2199   2250   2357   2408   2514   2565
                         2817   2913   3053   3170   3277
$OFILL  037367           3532*  3536*  3546   3588#
$40CAT= ****** U         3302
.       = 111054          547#   551#   558    559#   561#   563#   565#   567#   576    577#   579#   581#   599#
                          718    834#  1021#  1113   1149   1178   1288   1299   1306   1310   1332   1361   1375
                         1379   1396   1410   1414   1432   1458   1476   1480   1497   1528   1572   1596   1626
                         1656   1686   1720   1950   2330   2488   2645   3000   3003   3139   3143   3189#  3279
                         3280#  3336   3370#  3501#  3659#  4412#  4431#  4432#  4433#
.IIST   = 000000          478#  3164
.$ASTA= ****** U         3453   3456
.$X     - 000230          576#   581
```

```
AFORK      18#  1865  1968  2037
BFORK      19#  2660
BLK       602#   611   615   619   623   636   640   644   648   652   661   670   674   712
BLKB      602#   603   607   628   632   656
COMMEN      1#   472#
DEQUE      28#  3715  3737
E           1#
ENDCOM      1#   472#
ENQUE      26#  3688  3694
ERROR      95#  1919  1941  1956  2133  2169  2224  2300  2317  2382  2458  2475  2539  2615  2632
         2765  2824  2834  3815
ESCAPE      1#   472#
EXTRAS     32#   787
GETLID     26#  1787  3281
GETPID     25#  1945
GETPRI      1#   472#
GETSID     24#
GETSWR      1#   472#
IISTD      23#
IISTE      22#  1929
LOOP       15#  1312  1349  1381  1416  1444  1482  1511  1556  1581  1611  1641  1671  1701  1735
LOOPX      15#  1312  1349  1382  1416  1444  1482  1511  1557  1581  1611  1641  1671  1701  1735
MEMBOX     20#  2184  2342  2499
MULT        1#   472#
NEWTST      1#   472#  1262  1316  1351  1386  1421  1446  1486  1515  1561  1585  1615  1645  1675
         1705  1859  1963  2032  2178  2336  2493  2651
P          16#  3395
POP         1#   472#  3495  3496  3644
PUSH        1#   472#  3456  3458  3479  3603
QINIT      28#  1050
REPORT      1#   472#
RESPRI     30#
SAVPRI     29#
SCOPE      96#
SETK       15#  1283  1292  1321  1426  1451  1462  1491  1502  1520  1532  1566  1603  1633  1663
         1693  1710
SETPRI      1#   472#
SETS       15#  1356  1365  1590  1601  1620  1631
SETTRA   3790#  3799  3800  3801  3802
SETU       15#  1391  1400  1650  1661  1680  1691
SETUP       1#   472#
SKIP        1#   472#
SLASH       1#   472#
SPACE     472#
STARS       1#   472#   556   573   575   582   595   718   721  1262  1264  1316  1318  1351  1353
         1386  1388  1421  1423  1446  1448  1486  1488  1515  1517  1561  1563  1585  1587  1615
         1617  1645  1647  1675  1677  1705  1707  1859  1861  1963  1965  2032  2034  2178  2180
         2336  2338  2493  2495  2651  2653  3146  3292  3451  3508  3593  3766
SWRSU       1#   472#
SYNC       20#
TEST       15#  1265  1319  1354  1389  1424  1449  1489  1518  1564  1588  1618  1648  1678  1708
         1862  1966  2035  2181  2339  2496  2654
TRMTRP   3790#
TSTNO      20#  1898  2000  2071  2232  2390  2547  2665
TSTNUM     15#  1266  1320  1355  1390  1425  1450  1490  1519  1565  1589  1619  1649  1679  1709
         1863  1967  2036  2182  2340  2497  2655
```

```
TYPBIN      1#    472#
TYPDEC      1#    472#  1139  1823  1907  2009  2089  2100  3190  3206  3215  3224  3233  3240
TYPNAM      1#    472#
TYPNUM      1#    472#
TYPOCS      1#    472#
TYPOCT      1#    472#  1089
TYPTXT      1#    472#  3186  3193  3200  3209  3218  3227  3236  3433
UBER       32#   1438   1465  1504  1534  1545  1604  1634  1664  1694  1726  2907
UBES       31#   1285   1329  1358  1393  1429  1455  1494  1525  1569  1593  1623  1653  1683  1717
V          17#   3440
VS         18#
$MSTRT      1#
$$CMRE    593#    696    698   700   702   704
$$CMTM    593#    706    707   708   709   710
$$ESCA      1#    472#
$$NEWT      1#    472#  1262  1316  1351  1386  1421  1446  1486  1515  1561  1585  1615  1645  1675
          1705    1859  1963  2032  2178  2336  2493  2651
$$SET     3790#   3799  3800  3801  3802
$$SKIP      1#    472#
.EQUAT      1#
.HEADE      1#
.IIST       1#    475
.KT11       1#
.SETUP      1#    571
.SWRHI      1#
.$ACT1      1#    554
.$APTB      1#    719#
.$APTH      1#    571
.$APTY      1#   3449
.$ASTA      1#
.$CATC      1#    545
.$CMTA      1#    593
.$DB2D      1#
.$DB2O      1#
.$DIV       1#
.$EOP       1#   3144
.$ERRO      1#
.$ERRT      1#
.$MULT      1#
.$POWE      1#
.$RAND      1#
.$RDDE      1#
.$RDOC      1#
.$READ      1#
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#   3290
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#   3764
.$TYPB      1#
.$TYPD      1#   3591
.$TYPE      1#
.$TYPO      1#   3506
```

```
.$40CA      1#
.1170       1#      88


. ABS.  111054      000


 ERRORS DETECTED:  0

 DSKZ:CEKBGB.BIN,DSKZ:CEKBGB.LST/CRF/SOL-CEKBGB.SML,CEKBGB.P11
 RUN-TIME: 47 60 5 SECONDS
 RUN-TIME RATIO: 325/114=2.8
 CORE USED:  39K  (77 PAGES)
```