HP 3000 Computer Systems

# MICRO 3000 SELFTEST and MAINTENANCE MODE

Diagnostic Manual

HEWLETT PACKARD
8010 FOOTHILLS BLVD., ROSEVILLE, CA 95678

Part No. 30534-90001
E1286

Printed in U.S.A. 12/86

# PRINTING HISTORY

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The dates on the title page change only when a new edition or a new update is published. No information is incorporated into a reprinting unless it appears as a prior update; the edition does not change when an update is incorporated.

First Edition . . . . . Dec 1986

# LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the date of the most recent version of each page in the manual. To verify that your manual contains the most current information, check the dates printed at the bottom of each page with those listed below. The date on the bottom of each page reflects the edition or subsequent update in which that page was printed.

| Effective Pages | Date |
|---|---|
| all | Dec 1986 |

# CONTENTS

# CONTENTS (continued)

# CONTENTS (continued)

# FIGURES AND TABLES

## LIST OF TABLES



## LIST OF FIGURES

# ERROR CODES

APPENDIX A

Appendix A includes all error codes associated with the MICRO 3000 selftest. The error codes are listed according to the test sequence they are associated with: ATP, PIC, LANIC, CPU, or MEMORY.

## ATP TEST ERROR CODES

The ATP error codes are divided according to one of the nine sections of the ATP test that detected the error.

The slot test will display one of the following error codes if the ATP test fails. The tests are run in the same sequence as the test sections. Testing stops when the first failure is detected.

### ATP Test Section 1

This section is the initialization test which initializes the ATP and verifies registers contain expected data.

| Code | Error |
|---|---|
| 0108 | Register 8 initialization error (expected $0800) |
| 0109 | Register 9 initialization error (expected $0800) |
| 010A | Register A initialization error (expected $FF00) |
| 010C | Register C initialization error (expected $0000) |
| 010D | Register D initialization error (expected $0000) |
| 010E | Register E initialization error (expected $5004) |
| 010F | Register F initialization error (expected $5004) |

### ATP Test Section 2

This section tests basic operations, and OBII, IPOLL, SMSK and RMSK.

| Code | Error |
|---|---|
| 0201 | SMSK/RMSK test; ATP did not respond with mask bit set. |
| 0202 | SMSK/RMSK test; ATP did not respond with mask bit clear. |
| 0203 | ATP did not set IRQ. |
| 0204 | Improper IPOLL response (none, wrong, or multiple channels). |
| 0205 | Improper OBII response. |

## ATP Test Section 3

The section is the port register test which tests registers 0-7 on ports 0-7.

| Code | Error |
|---|---|
| 0300 | Port 0, register 0 pattern test failure |
| 0301 | Port 0, register 1 pattern test failure |
| . | |
| . | |
| 0307 | Port 0, register 7 pattern test failure |
| 0310 | Port 1, register 0 pattern test failure |
| . | |
| . | |
| 0377 | Port 7, register 7 pattern test failure |

## ATP Test Section 4

This section is the DMA test. SIMB write Word.

| Code | Error |
|---|---|
| 0401 | DMA state machine failed to go to state 4. |
| 0402 | DMA state machine failed to go to state 3. |
| 0403 | DMA write to memory transferred improper data. |
| 0404 | DMA read from memory to RBYTE transferred improper data. |
| 0405 | DMA read from memory to LBYTE transferred improper data. |
| 0406 | DMA counter test failed. |

## ATP Test Section 5

Port selftest.

| Code | Error |
|---|---|
| 0500 | Port 0 selftest failure |
| 0501 | Port 1 selftest failure |
| . | |
| . | |
| 0507 | Port 7 selftest failure |

## ATP Test Section 6

This section is the DMA loopback to port test.

| Code | Error |
|---|---|
| 0600 | Port 0 loopback failure |
| 0601 | Port 1 loopback failure |
| . | |
| . | |
| 0607 | Port 7 loopback failure |

## ATP Test Section 7

This section is the DMA loopback data test.

| Code | Error |
|---|---|
| 0700 | Port 0 loopback data failure |
| 0701 | Port 1 loopback data failure |
| . | |
| . | |
| 0707 | Port 7 loopback data failure |

# PIC TEST ERROR CODES

If the PIC test fails, the slot test will display one of the following error codes. The tests are run in the same sequence as the test sections. Testing stops when the first failure is detected.

## PIC Test Section 1

This section is the initialization test which initializes the PIC and verifies registers contain expected data.

| Code | Error |
|---|---|
| 0104 | Register 4 initialization error (expected $0000) |
| 0105 | Register 5 initialization error (expected $0000) |
| 0106 | Register 6 initialization error (expected $0020) |
| 0107 | Register 7 initialization error (expected $0000) |
| 0108 | Register 8 initialization error (expected $0000) |
| 010A | Register A initialization error (expected $0000) |
| 010C | Register C initialization error (expected $0000) |
| 010F | Register F initialization error (expected $0087+(8*channel#)) |

## PIC Test Section 2

This section tests basic operations and OBII, IPOLL, SMSK and RMSK.

| Code | Error |
|---|---|
| 0201 | SMSK/RMSK test; PIC did not respond with mask bit set. |
| 0202 | SMSK/RMSK test; PIC did not respond with mask bit cleared. |
| 0203 | PIC did not set IRQ after SMSK on selected channel. |
| 0204 | Improper IPOLL response after SMSK (no response, wrong response, or multiple channels responding). |
| 0205 | Improper OBII data from this channel after SMSK. |
| 0206 | Improper register D response after SMSK. |
| 0207 | Improper IPOLL response (should have been clear). |
| 0208 | Received no CSRQ after issuing a SIOP. |
| 0209 | Wrong channel is responding to SPOLL. |
| 020A | Improper OBSI data after SPOLL. |
| 020B | Improper register F data after SPOLL. |
| 020C | Improper CSRQ response after HIOP (should have been zero). |
| 020D | Improper SPOLL after HIOP (should have been zero). |
| 020E | Improper OBSI response to SPOLL after HIOP (should have pointed to device 7). |

## PIC Test Section 3

This section tests the HB-IB controller chip Status Register (PIC Register 1).

| Code | Error |
|---|---|
| 0301 | Initialization error - bits 0 to 7 should be zero. |
| 0302 | Initialization error - bits 10, 13, and 14 should be zero and bits 11 and 12 should be one. |
| 0303 | Bit 13 should be 1, bit 14 should be zero when the HP-IB controller chip is addressed to talk, but not listen. |
| 0304 | Bits 13 and 14 should both be one when the HP-IB controller chip is addressed to talk and listen. |
| 0305 | Bit 14 should be one, bit 13 should be zero when the HP-IB controller chip is addressed to listen, but not to talk. |
| 0306 | Bits 13 and 14 should both be zero when the HP-IB controller chip is not addressed to talk or to listen. |

## PIC Test Section 4

This section tests the HP-IB controller chip Interrupt Register (PIC Register 2) and Interrupt Mask Register (PIC Register 3).

| Code | Error |
|---|---|
| 0401 | Reg2, bit 0 should be set (an interrupt is pending). |
| 0402 | Reg2, bits 9 and 13 should be clear (no handshake abort and inbound FIFO empty). |
| 0403 | Reg2, bit 12 should be set (outbound FIFO room available). |
| 0404 | Reg2, bit 14 should be set (outbound FIFO idle). |
| 0405 | Reg2, bit 8 should be clear (no status change). |
| 0406 | Reg2, bit 0 should be clear (no interrupt pending). |
| 0407 | Reg2, bit 8 should be set (status change occurred). |
| 0408 | Reg2, bit 9 should be clear (no handshake abort). |
| 0409 | Reg2, bits 12 and 14 should be clear. Bit 13 should be set (outbound FIFO room not available, outbound FIFO not idle, and inbound FIFO not empty. ==> FIFO bits in opposite state). |
| 040A | Reg2, bit 9 should be set (handshake abort). |
| 040B | Reg2, bit 9 should be clear (no handshake abort). |

## PIC Test Section 5

This section tests PIC registers 4 and 5 by reading and writing data. Functions are not tested.

| Code | Error |
|---|---|
| 0501 | Register 4 fails to show data patterns $00e. |
| 0502 | Register 5 fails to show data patterns $00AA. |
| 0503 | Register 4 fails to show data patterns $0055. |
| 0504 | Register 5 fails to show data patterns $0055. |

## PIC Test Section 6

This section tests PIC register 6 (HP-IB control register).

| Code | Error |
|---|---|
| 0601 | Register 6 fails to show patterns $802A for read/write test. |
| 0602 | Register 6 fails to show patterns $4054 for read/write test. |
| 0603 | Bits 12 and 14 of the HP-IB controller chip interrupt register (PIC register 2) should be set (outbound FIFO NOT full and idle. |
| 0604 | Bits 12 and 14 of the HP-IB controller chip interrupt register (PIC register 2) should be clear (outbound FIFO NOT full and idle. |
| 0605 | Bit 15 of register 6 should be set (clear outbound FIFO) via the HP-IB controller chip interrupt register test. Bits 12 and 14 of the HP-IB controller chip interrupt register should be clear. |

## PIC Test Section 7

This section tests PIC register 7 (HP-IB address).

| Code | Error |
|---|---|
| 0701 | Register 7 fails to show patterns $800A for read/write test. |
| 0702 | Register 7 fails to show patterns $4015 for read/write test. |
| 0703 | Bit 9 of Register 7 should be set (talk always) via the HP-IB controller chip status register (PIC register 1) test. Bits 13 or 14 of the HP-IB controller chip register should be off to indicate NOT a talk or a listen. |
| 0704 | Bit 10 of register 7 should be set (listen always) via the HP-IB controller chip status register test. Bit 13 of the HP-IB controller chip status register must be on to indicate a talk always. |

| | |
|---|---|
| 0705 | Same as 0704 except bit 14 of the HP-IB controller chip status register must be off to indicate NOT a listen always. |
| 0706 | Same as 0704 except the HP-IB controller chip status register is again updated and bit 13 of the HP-IB controller chip status register must be on to indicate NOT a listen always. to indicate NOT a listen always. |

## PIC Test Section 8

This section tests PIC register 8, 9 and 10 with read/write data only. Register 8 is tested for lower 8 bits only. Register 9 and 10 ($A) are tested for all 16 bits.

| Code | Error |
|---|---|
| 0801 | Register 8 fails to show data patterns $00AA. |
| 0802 | Register 9 fails to show data patterns $AAAA. |
| 0803 | Register 10 fails to show data patterns $AAAA. |
| 0804 | Register 8 fails to show data patterns $0055. |
| 0805 | Register 9 fails to show data patterns $5555. |
| 0806 | Register 10 fails to show data patterns $5555. |

## PIC Test Section 9

This section tests DMA Write/Read/Abort from memory to PIC FIFO, and from PIC FIFO to memory.

| Code | Error |
|---|---|
| 0901 | CSRQ response & test via OBSI failed the DMA write to PIC. |
| 0902 | Data transferred to the PIC FIFO by the above transfer (assuming the DMA write abort test passed) is incorrect. |
| 0903 | Bit 9 of the interrupt register (PIC reg 2) should be clear (no handshake abort) the DMA write abort (PIC reg E) test. |
| 0904 | CSRQ response & test via OBSI failed the DMA write abort. |
| 0905 | Bit 5 of PIC register B should be set for the DMA write abort termination test. |
| 0906 | Bit 6 of PIC register B should be set for the DMA write abort termination test. |
| 0907 | CSRQ response & test via OBSI failed the DMA read from PIC. |
| 0908 | Data read from PIC on DMA read test different than expected. |
| 090A | CSRQ response & test via OBSI failed for the DMA read abort. |
| 090B | Bit 5 of PIC register B should be set for the DMA read abort termination test. |
| 090C | Bit 6 of PIC register B should be set for the DMA read abort termination test. |
| 090F | Timeout. Waiting for CSRQ. |

# LANIC TEST ERROR CODES

## LANIC Selftest Subtest Codes

LANIC selftest subtest descriptions and codes are listed below:

| Code | Test |
|---|---|
| 0001 | Z80 instruction set |
| 0002 | EPROM checksum |
| 0003 | Station address PROM checksum |
| 0004 | High byte latch |
| 0005 | RAM data (background test) |
| 0005 | Byte RAM data (even addresses) |
| 0006 | Byte RAM data (odd addresses) |
| 0007 | Byte RAM address (incrementing addresses) |
| 0008 | Byte RAM address (decrementing addresses) |
| 0009 | Word RAM address |
| 000A | Word/Byte address mapping |
| 000B | Z80 memory reference instructions |
| 000C | Values from reset MDIAG,SYSCON |
| 000D | CTC data test |
| 000E | CTC mode 0 counting |
| 000F | CTC mode 2 counting |
| 0010 | CTC mode 4 counting |
| 0011 | Interrupt PAL (bit 4) |
| 0012 | Z80 interrupt |
| 0013 | Z80 non-maskable interrupt |
| 0014 | Master handshake disabled (MHSDIS) |
| 0015 | PADDR TO BADDR (low 15 bits) |
| 0016 | ZBANKL |
| 0017 | ZBANKH |
| 0018 | Preliminary FIFO (INREADY, ADVREADY, OUTREADY) |
| 0019 | FIFO data (BDATA 7) |
| 001A | FIFO data (BEA 7,8) |
| 001B | FIFO data (BDATA 2:6) |
| 001C | FIFO data (BDATA 0,1,13:15) |
| 001D | FIFO data (BDATA (8:12) |
| 001E | FIFO data (BA 11:15) |
| 001F | FIFO data (BA 6:10) |

| | |
|---|---|
| 0020 | FIFO data (BA 1:5) |
| 0021 | R14 configuration register |
| 0022 | OBII value; channel number not 0 |
| 0023 | COMCON values from reset |
| 0024 | MAU power on/off |
| 0025 | R13 CR, CR full bit |
| 0026 | R15 selftest result register |
| 0027 | 82586 interrupt |
| 0028 | 82586 reset |
| 0029 | Register decode (PBUS register addressing) |
| 002A | 82586 LANIC RAM addressing |
| 002B | 82586 diagnose command |
| 002C | Level 1 chip loopback |
| 002D | 82586 write to FIFOS |
| 002E | MAU loopback on media |

## LANIC Selftest Error Codes

The system console will display one of the following error codes if the LANIC test fails:

| Code | Error |
|---|---|
| 0179 | Failure while performing 82586 initialization for the LANIC diagnostic. |
| 017A | The 82586 did not clear its command word prior to interrupting. |
| 017B | Selftest result register (R15) bit is bad. |
| 017C | Z80 stack underflow during selftest. |
| 017D | Unexpected Z80 non-maskable interrupt (NMI). |
| 017E | Unexpected Z80 interrupt. |
| 017F | LANIC was reset but selftest never started, or LED circuitry failed. (This code is set in the LEDs by the hard reset preceding every full selftest.) In this case, the selftest may have a failure code or a random value. |

## CPU TEST ERROR CODES

The system console displays one of the following error codes if the CPU test fails. The tests are run in the same sequence as the test sections. Testing stops when the first failure is detected.

Test Section 1 - Bank register tests

0101 Pbank read/Abank write failure
0102 Dbank read/Sbank write failure
0103 Sbank read/Dbank write failure
0104 Abank read/Pbank write failure

Test Section 2 - TOC RAM tests

0201 TOC RAM test not done (power fail)
0202 TOC RAM data failure

Test Section 3 - TOC count verification

0203 TOC not counting

Test Section 4 - MPE timer verification

0301 TOC not counting

Test Section 5 - Watchdog timer verification

0501 Watchdog timer did not rollover

## MEMORY TEST ERROR CODES

The system console displays one of the following error codes if the memory test fails. The tests are run in the same sequence as the test sections. Testing stops when the first failure is detected.

| Code | Error |
|---|---|
| 0000 | PFAR test failed (memory dead) |
| 0001 | Memory size test failed |
| 0002 | Memory initialization test failed |
| 0003 | Memory address test failed |
| 0004 | Memory pattern test failed |
| 0005 | Parity test failed |

# POWER-ON SELFTEST LED INTERPRETATION

Table A-1. Power-On Selftest LED Interpretation

| ACTIVITY LED | FAULT LED | LED INTERPRETATION | ACTION TO TAKE |
|---|---|---|---|
| OFF | OFF | No Power.<br><br>Processor completely dead. | Check for AC. Measure DC supply output.<br><br>Replace processor. |
| ON | ON | Executing selftest.<br><br>System in battery back-up mode. | Wait for selftest completion.<br><br>Wait for AC power return. |
| OFF | ON | CPU or 0-2 Mb memory failed. | Replace processor PCA. |
| OFF | Flashing single blink | Unsupported ATP detected. | Remove unsupported ATP. |
| OFF | Flashing double blink | Console ATP failed.<br><br>Fast/slow SIMB I/F failed. | Replace ATP in slot 1 or replace processor PCA. |
| OFF | Flashing triple blink | Console failed to speed sense. | Check cable connection between console and ATP.<br><br>Check that console is in REMOTE mode. |
| OFF | Flashing triple blink | Console failed to speed sense. | Console may be defective. Attempt loopback datacomm test.<br><br>Replace ATP. |
| ON | OFF | Selftest execution complete.<br><br>"H for help" prompt displayed. | |

# TEST MODE COMMAND DESCRIPTIONS

SECTION 4

## INTRODUCTION

This section provides a definition of test mode and a description of test mode commands.

Test mode allows the system operator to initiate and manually direct the selftests. Test mode allows for looping up to 9999 times; the default is 1. Looping is disabled when the keyswitch is in position "1" (NORMAL).

Use the maintenance mode TE[st] command to enter test mode.

Error messages can be found in Appendix A.

## TEST MODE COMMANDS

Test mode allows you to enter the following commands at the "Test ->" prompt:

| | |
|---|---|
| All | Help |
| Channel | IOMAP |
| CPU | Memory |
| Exit | PON |

### All

The ALL command runs all of the manually directed selftests except the PON test in the following order:

- CPU test
- Memory test
- Channel test
- IOMAP

The test may be looped by specifying the desired number of loops in *count*. *Count* must be an integer between 1 and 9999; the default is 1. (A space is required before specifying *count*.)

The correct syntax for this command is:

```
AL[L] [ count]
```

The following illustrates the use of the ALL command:

```
1-NORMAL
Test ->AL

 TOC RAM
Addr Data
000E 0000
000F 0000
0010 0000
0011 0000
0012 0004
0013 000E
0014 0000
0015 0000
CPU Test passed
Memory Test passed
Channel  1  - Terminal Interface Controller
Channel  4  - Peripheral Interface Controller
Test Passed
                    System I/O Configuration
------------------------------------------------------------
Memory Size (MEGABYTES) = nn
Load:
Channel  4 Device 3
Start/Dump:
Channel  4 Device 1
------------------------------------------------------------
Channel  1  ID=4  - Terminal Interface Controller
------------------------------------------------------------
Channel  4  ID=2  - Peripheral Interface Controller
  Device 3  ID=0260  - 9144 Cartridge Tape Unit
  Device 1  ID=$022B  -  7958 Disc Drive Unit
------------------------------------------------------------

1-NORMAL
Test ->
```

## Channel

The CHANNEL test performs the ATP, PIC, and LANIC tests. The CHANNEL test may be looped by specifying the desired number of loops in *count*. *Count* must be an integer between 1 and 9999; the default is 1. (A space is required before specifying *count*.) *Count* must be specified whenever a specific channel is selected with the *channel* command. If a particular channel is not specified, all channels are tested. The appropriate test (ATP, PIC, LANIC or none) is run for each PCA installed in the CPU. If a failure occurs, the failure code is displayed on the system console next to the PCA description. Refer to Appendix A for ATP, PIC, and LANIC test error codes.

**ATP Test**

The ATP test has six sections:

1. Init check. Performs an ATP initialization and tests to verify registers contain proper data.
2. Basic I/O operations. Issues OBII, IPOLL, SMSK, and RMSK and verifies a proper response.
3. Port register tests. Writes patterns to registers 0-7 of ports 0-7 and verifies the data.
4. Diagnostic loopback using DMA sequencer ROM.
5. Initiates PCC tests on all 8 ports.
6. Performs DMA data loopback test on all 8 ports.

The console ATP is speed sensed and communication lines are tested with the local console.

**PIC Test**

The PIC test has nine sections:

1. Init check. Performs PIC initialization and verifies registers contain proper data.
2. Basic I/O operations. Issues OBII, IPOLL, SMSK, and RMSK and verifies a proper response.
3. Tests the HP-IB controller chip status register (PIC register 1).
4. Tests the HP-IB controller chip interrupt (PIC register 2) and interrupt mask registers (PIC register 3).
5. Tests PIC registers 4 and 5 using data patterns.
6. Tests PIC register 6.
7. Tests PIC register 7.
8. Tests PIC registers 8, 9, and A using data patterns.
9. Fills HP-IB controller chip FIFO and executes DMA to and from memory. Tests PIC registers associated with DMA transfers.

LANIC Test

The LANIC code is not included in the selftest microcode. The LANIC PCA executes its own selftest.

After the ATP, PIC, and LANIC tests execute, return is to the "Test ->" prompt.

The correct syntax for this command is:

CH[an][ *count*[,*channel*] ]

The following illustrates the use of CHAN:

```
1-NORMAL
Test ->CH
~
Channel  1  - Terminal Interface Controller
Channel  4  - Peripheral Interface Controller
Test Passed

1-NORMAL
Test ->
```

## CPU

The test mode CPU command executes the following CPU tests not run at power-on:

- P, D, S, and A bank register tests.
- TOC RAM location test.
- TOC counting verification.
- MPE timer counting verification.
- Watchdog Timer Force Condition verification and FMD capability test.

Tests not performed by this CPU test, but executed by the power-on CPU test are:

- ROM checksum test.
- Full processor chip function test.
- Full WCS address and data test.
- Register file address and data test.

The test may be looped by specifying the desired number of loops in *count*. *Count* must be an integer between 1 and 9999; the default is 1. (A space is required before specifying *count*.) The correct syntax for this command is:

```
CPU [ count]
```

The following illustrates the use of CPU:

```
1-NORMAL
Test ->CP

 TOC RAM
Addr Data
000E 0000
000F 0000
0010 0000
0011 0000
0012 0004
0013 000E
0014 0000
0015 0000
CPU test passed

1-NORMAL
Test ->
```

## Exit

The test mode EXIT command returns execution to maintenance mode and displays the "H for help->" prompt.

The correct syntax for this command is:

```
E[xit]
```

The following illustrates the use of EXIT:

```
Test ->E
H for help->
```

## Help

The HELP command does not appear in the test mode menu. When issued, the HELP command displays the available test mode commands and the ROM version number.

The correct syntax for this command is:

```
HELP
```

The following illustrates the use of HELP:

```
1-NORMAL
Test ->H


ROM Version:  nnnn

Selftest Menu:

AL[l] [ count ]
CH[an] [ count [ ,chan ] ]
CP[u] [ count ]
E[xit]
I[omap] [ count ]
M[emory] [ count ]
PON [ count ]


1-NORMAL
Test ->
```

## IOMAP

The IOMAP command executes a version of IOMAP contained in the selftest ROM. This version of IOMAP runs the memory size portion of the memory test, displays the number of megabytes installed in the system, lists the load and start/dump devices, and identifies all PCAs installed in the system. All supported HP-IB devices attached to the PIC are identified and their ID code is displayed with a device description.

The test may be looped by specifying the desired number of loops in *count*. *Count* must be an integer between 1 and 9999; the default is 1. (A space is required before specifying *count*.)

The correct syntax for this command is:

```
I[omap] [ count]
```

The following illustrates the use of IOMAP:

```
1-NORMAL
Test ->I

                    System I/O Configuration
------------------------------------------------------------
Memory Size (MEGABYTES) = nn
Load:
Channel  4 Device 3
Start/Dump:
Channel  4 Device 1
------------------------------------------------------------
Channel  1  ID=4  - Terminal Interface Controller
------------------------------------------------------------
Channel  4  ID=2  - Peripheral Interface Controller
  Device 3  ID=$0260  - 9144 Cartridge Tape Unit
------------------------------------------------------------

1-NORMAL
Test ->
```

## Memory

The test mode MEMORY command executes the power-on memory test. The amount of installed memory is determined, then memory is initialized before a refresh test, an address test, a pattern test and a parity test are performed. Upon test completion, memory is left with $30F8 (halt 8) in all locations.

The full memory test executes. If an error is detected, the system console displays the memory test section number the error was detected in. The FAULT LED will light and the ACTIVITY LED will go out. Refer to Appendix A for list of memory test error codes.

The test may be looped by specifying the desired number of loops in *count*. *Count* must be an integer between 1 and 9999; the default is 1. (A space is required before specifying *count*.)

Return is to the "Test ->" mode prompt.

The correct syntax for this command is:

```
M[emory] [ count]
```

The following illustrates the use of MEMORY:

```
1-NORMAL
Test ->M

Memory Test passed
```

## PON

The test mode PON command executes the power-on selftest *count* times. This command is like the ALL test mode command except the power-on CPU test is executed in place of the manually executed CPU test, and IOMAP is not executed. The PON test is initiated by toggling the PON line.

**NOTE**

This command cannot be run from keyswitch position "3" (REMOTE) or from keyswitch position "2" (LOCAL) directly from position "3" (REMOTE).

The correct syntax for this command is:

PON [*count*]

The following illustrates the use of PON:

```
2-LOCAL (from Normal)
Self Test ->PON
~
Power on Self Test
Memory Test passed
Memory Size (MEGABYTES) = nn
Channel  1  - Terminal Interface Controller
Channel  4  - Peripheral Interface Controller

2-LOCAL (from Normal)
Self Test ->
```

# SOFTPANEL

SECTION 5

## INTRODUCTION

Softpanel is a diagnostic tool used to examine software. Softpanel allows the user to display and modify memory, perform register and I/O operations, and perform other necessary functions. All commands requiring parameters must have a "+", "-", or a space between the the command and the parameter. The following commands are allowed in softpanel:

Display Memory

Modify Memory

Register Operations
- DR
- MR

Execution Control
- E
- RUN

Input/Output Operations
- RIO
- WIO

Other/Miscellaneous
- T
- ENV
- RTOC
- WTOC
- RDX
- ST

## COMMAND PARAMETERS

Softpanel command parameters are defined as:

*bank* — One of the following numeric fields limited to a range of 0 - 255 (8 bits): the current radix numeric field, a hexadecimal numeric field preceded by a $ or a digit, or an octal numeric field preceded by a %.

*count* — One of the following numeric fields limited to a 16 bit maximum: the current radix numeric field, a hexadecimal numeric field preceded by a $ or a digit, or an octal numeric field preceded by a %.

*expr* — A combination of *numeric* and *op*. Operations are performed from left to right with no precedence.

*ioaddr* — One of the following numeric fields limited to a 16 bit maximum: the current radix numeric field, a hexadecimal numeric field preceded by a $ or a digit, or an octal numeric field preceded by a %.

| | |
|---|---|
| *iodata* | One of the following numeric fields limited to a 16 bit maximum: the current radix numeric field, a hexadecimal numeric field preceded by a $ or a digit, or an octal numeric field preceded by a %. |
| *numeric* | One of the following numeric fields limited to a 16 bit maximum: the current radix numeric field, a hexadecimal numeric field preceded by a $ or a digit, or an octal numeric field preceded by a %. |
| *op* | One of: a +, -, or *. Operations done on numeric fields are signed. Bit (0:1) is the sign bit. : is the indirection operator. |
| *reg* | One of the following registers: DB, DL, Q, S, PB, PL, Z, STA (the HP 3000 status register), SB (the split bank flag, 1 bit wide), CIR, X, SW (the switch register containing load/boot device DRT), Dbank, Sbank, Pbank, LPFlg, DISP, or ICS. |
| *regfile* | One of the following numeric fields limited to a range of 0 - 255 (8 bits): the current radix numeric field, a hexadecimal numeric field preceded by a $ or a digit, or an octal numeric field preceded by a %. |
| *tocaddr* | One of the following numeric fields limited to a 16 bit maximum: the current radix numeric field, a hexadecimal numeric field preceded by a $ or a digit, or an octal numeric field preceded by a %. |
| *tocdata* | One of the following numeric fields limited to a 16 bit maximum: the current radix numeric field, a hexadecimal numeric field preceded by a $ or a digit, or an octal numeric field preceded by a %. |

# COMMAND DESCRIPTIONS

## Display Memory

The D command continues from the last screen displayed and displays another half screen of data. All display commands display in the current radix (refer to SDM [Set Display Mode] command). The display command will always show multiples of 8 words in the current radix and in ASCII.

The options are:

| | |
|---|---|
| DA *expr*[:[[+]/[-]]*expr*]][,*count*] | Displays memory at the given absolute address in bank 0. |
| DEA *bank.expr*[:[[+]/[-]]*expr*]][,*count*] | Displays the absolute address relative to the specified bank. |
| DSY[[+]/[-]/[ ][*expr*]][:[[+]/[-]]*expr*]][,*count*] | Displays the absolute address relative to sysglobal. |

DDB[+ -expr [ [expr]]][:[+ -]expr]][,count] Displays the absolute address relative to the DB register.

DDL[+ -expr [ [expr]]][:[+ -]expr]][,count] Displays the absolute address relative to the DL register.

DQ [+ -expr [ [expr]]][:[+ -]expr]][,count] Displays the absolute address relative to the Q register.

DPB[+ -expr [ [expr]]][:[+ -]expr]][,count] Displays the absolute address relative to the PB register.

DS [+ -expr [ [expr]]][:[+ -]expr]][,count] Displays the absolute address relative to the S register.

DZ [+ -expr [ [expr]]][:[+ -]expr]][,count] Displays the absolute address relative to the Z register.

DP [+ -expr [ [expr]]][:[+ -]expr]][,count] Displays the absolute address relative to the P register.

DPL[+ -expr [ [expr]]][:[+ -]expr]][,count] Displays the absolute address relative to the PL register.

## Modify Memory

Modify memory commands will display the current address, current contents, and wait for the user to input a new value. Input the new value using a numeric field with the current default radix or force the new value using the radix forces ("%" or "$"). The command will terminate when the user inputs either "." or "//" in response to the prompt.

The options are:

MA *expr*[:[+ -]*expr*] Modify the absolute address in bank 0.

MEA *bank.expr*[:[+ -]*expr*] Modify the absolute address in the specified bank.

MSY[+ -*expr* [ [*expr*]]][:[+ -]*expr*] Modify the absolute address in sysglobal.

`MDB[+|-expr|] [:[+|-expr]]` Modify the absolute address in the DB register.

`MDL[+|-expr|] [:[+|-expr]]` Modify the absolute address in the DL register.

`MQ [+|-expr|] [:[+|-expr]]` Modify the absolute address in the Q register.

`MS [+|-expr|] [:[+|-expr]]` Modify the absolute address in the S register.

`MZ [+|-expr|] [:[+|-expr]]` Modify the absolute address in the Z register.

`MPB[+|-expr|] [:[+|-expr]]` Modify the absolute address in the MPB register.

`MP [+|-expr|] [:[+|-expr]]` Modify the absolute address in the P register.

`MPL[+|-expr|] [:[+|-expr]]` Modify the absolute address in the PL register.

## Register Operations

The DR command will display the common registers (i.e., P, PB, PL, CIR, DB, Q, S, etc.). If no field is specified to DR then all common registers will be displayed.

The options are:

`DR [reg | regfile[,count]]` Display the value contained in a register.

`MR {reg | regfile}` Modify the value contained in a register.

## Execution Control

The options are :

E Exit back to maintenance mode

RUN Run (return to software)

## Input/Output Operations

The addresses and data patterns for Input/Output operations conform to IMB and SIMB formats.

The options are:

RIO *ioaddr* Read I/O from address *ioaddr*

WIO *ioaddr,iodata* Write I/O address *ioaddr* with data *iodata*

## Miscellaneous Commands

The T (Trace) command allows the user to trace the current (or specified) stack. ENV allows the user to move back markers of the current stack and access data there as if it were at the current marker. RDX allows the user to change the current radix. Softpanel starts with the radix set to octal. The ENV command specified with no parameters will turn ENV off (q-relative addresses revert to the current environment).

The options are:

T[ [*numeric.*]*numeric*] Trace stack.

ENV[ *numeric*] Change the environment.

RTOC *tocaddr* Read TOC RAM address.

WTOC *tocaddr,tocdata* Write TOC RAM with data.

RDX $\{^{H}_{O}\}$ Change the current radix. (*H* - Hex; *O* - Octal)

ST Give softpanel status.

## CPU ROMS Date Code 2647 Exceptions

1. In the softpanel: Any hexadecimal value starting with an alphabetic character (for example, "A","A00","C0") must be preceded by either a zero or a dollar sign, even if the current radix is set to hexadecimal. Imbedded letters in a hex value starting with a numeric character (for example, "1C00") do not have this restriction. The reason for this qualification for *numeric* in the hex radix is to distinguish between the DB register and the value "$DB".

2. In the softpanel, the T (Trace) command to a non-existent s-bank does not print an error message.

3. In the softpanel, the T (Trace) command will print the last user stack if the current stack is the ICS, but there is no way to use the ENV command to move back to the user stack.

4. In the softpanel, using the RIO command to a non-existent register produces a watchdog timer interrupt, and the console comes back to the maintenance mode prompt.

5. The switch register (SR) is not in the default register display. It may be displayed manually.

6. In the software display, the status register flags M, I, T, R, O, and C indicate a value of 1 with an upper case letter and a value of 0 with a lower case letter.

# TOC RAM INFORMATION

APPENDIX B

Appendix B contains tables defining the TOC RAM locations displayed on the system console during selftest execution.

## TOC RAM DATA AND STATUS TABLES

**Table B-1. TOC RAM Data and Status**

| TOC RAM ADDRESS | STATUS AND DATA STORED IN TOC RAM ADDRESS |
|---|---|
| $0A | Register A |
| $0B | Register B |
| $0C | Register C |
| $0D | Register D |
| $0E | Undefined |
| $0F | Console ATP port 0 interrupt time-out flag* |
| $10 | Last stop information. See Table B-2. |
| $11 | Start device |
| $12 | Load device |
| $13 | Undefined |
| $14 | Test loop counter (lower byte) |
| $15 | Test loop counter (upper byte) |
| $16-$3F | Undefined |

* If the ATP port 0 interrupt time-out flag is set (contains $AA) this means port 0 of the ATP in slot 1 of the SPU has failed to produce an expected interrupt within 500 milliseconds and the microcode has timed-out. This flag should be checked if the RUN command cannot be executed. If the flag is set, port 0 configuration information was not stored and thus could not be restored upon executing the RUN command.

Table B-2. Last-Stop Value Meanings

| VALUE | MEANING |
|---|---|
| $00-$0F | Halts 0 through 15 (halt instructions executed) |
| $10-$1F | System Halt 0 through 15 (firmware detected traps). See Table B-3. |
| $20 | WCS parity error |
| $21 | Watchdog Timer |
| $22 | Power failure |
| $23 | Control B (maintenance mode invoked) |
| $24 | Multiple bit parity error |
| $25 | SIMB bus parity error |
| $7F | MPE UP status removed by software. Disables Power Fail Auto Restart. |
| $80 | System was running MPE or other software. Enables Power Fail Auto Restart. |
| $81-$FF | Unused. These values may be seen upon TOC power-on. |

**NOTE**

Values between $00 and $FF not shown in Table B-2 are undefined.

**Table B-3. System Halt Causes**

| SYSTEM HALT | TOC VALUE | CAUSE |
|---|---|---|
| 1 | $11 | STT violation segment 1 |
| 2 | $12 | Absent trap while on ICS |
| 3 | $13 | Code segment 1 trap violation |
| 4 | $14 | ICS stack overflow |
| 6 | $16 | Initial program load failure |
| 7 | $17 | Illegal SBnk at QI-5 during IXIT |
| 9 | $19 | Psuedo-Enable when enabled |

**NOTE**

Values between $10 and $1F not shown in Table B-3 are undefined.