

Attached is a copy of the new CPBOOT functional specification. The specification describes how the CPBOOT program interfaces with systems that contain a diagnostic processor. Sense switch settings for the CPBoot routine can be found in appendixes A and B. If the system is running pre Rev 20 software the following error message will appear if switch 000002 is not set (Appendix B).

[CPBOOT Rev. 4.0 Copyright (c) Prime Computer, Inc. 1985]

[Failure Report]

Testing: MTBOOT Revision Compatability.
Actual: INCOMPATIBLE
Expected: COMPATIBLE

For pre-REV-20 tapes use:
BOOT <SS-value>2

Examples:
BOOT 15 2
BOOT 405 2
BOOT 505 2
BOOT 1205 2

NOTE: Controller '000014, Unit 000000 must be used.

DPM400: CPU halted at 012002: 005025
28 Aug 85 14:24:34 Wednesday

The CPBOOT routine has been incorporated into the 2655 and 9655 systems at the time of FCS. Other systems must have the following FCO's installed in order to run the new CPBOOT.

System Type	FCO Number	Part Number	Revision
9955	384	DSK7084-909	F
9950	382	DSK7084-901	T
9750	383	DSK7084-907	G
9650	387	DSK7084-902	K
2550	386	DSK7084-905	E
9655	N/A	DSK7084-915	A
2655	N/A	DSK7084-913	A

Table of Contents

1	Overview	4
2	References	4
3	CPBoot	5
4	Hardware Interface	5
	4.1 Required for Execution of CPBOOT	5
	4.2 Required for Successful Completion of CPBoot	5
	4.3 Required for Display of Failure Report	6
	4.4 Compatibility	6
5	Software Interface	6
6	Design Constraints	7
7	Assumptions	7
8	Differences	7
9	User Interface	7
10	Display Formats	8
	10.1 Informative Messages	8
	10.2 Error Messages	10
11	Performance	14
12	CPBOOT Algorithm	14
13	Hardware Verification	15
	13.1 Hard Core Instructions	15
	13.2 Checksum	15
	13.3 System Console	15
	13.4 Register File Test	15
	13.5 V-mode Addressing Test	16
	13.6 Memory Test	16
	13.7 Sanity Test	16
14	Boot Procedure	17
15	Appendix A	18
16	Appendix B	19
17	Packaging	20

1 Overview

The purpose of this specification is to describe the new CPBOOT program and its uses on the Prime products that incorporate a diagnostic processor. All references to the term "CPBoot" imply this new boot strap program.

The task of raising a central processor from a dormant state to an active state is the primary role of the CPBOOT program. CPBOOT provides the essential mechanism of transferring stored information from the source media to the target memory, thus providing the central processor with the necessary instructions to perform a useful task. The useful task, in most cases, is to perform the next boot strap routine.

Because CPBOOT is the only mechanism used for leaving the dormant state, it has the potential of being the weak line in the chain of events. In a worst case, the field engineer will attempt to use CPBOOT to load a diagnostic test to further diagnose a hardware failure. However, if the boot mechanism itself fails, the field engineer will not be able to load diagnostics. Therefore, the boot mechanism, CPBOOT, will provide diagnostic capability and will report boot failures in a friendly manner.

3 CPBOOT

CPBOOT is a boot strap program. It provides the linkage in a chain of boot strap routines. Upon power up, the diagnostic processor provides the first boot strap required to load CPBOOT. Next, CPBOOT provides the boot strap required to load another boot strap program from the selected I/O device. In addition to providing the boot strap mechanism, CPBOOT also provides CPU and I/O diagnostic coverage. This specification is limited to defining the diagnostic coverage and the boot procedure. For a detailed description of the diagnostic coverage, and the boot procedure, refer to the sections on "Hardware Verification" and "Boot Procedure". CPBOOT is written in PMA (Prime Macro Assembler) and uses the "hard core instruction subset" defined in PE-T-1055.

4 Hardware Interface

4.1 Required for execution of CPBOOT

Depending on the system, CPBOOT will reside on the diagnostic processor floppy disk or the diagnostic processor EPROM card. For execution of CPBoot to begin, the diagnostic processor bootstrap mechanism must be capable of transferring CPBOOT from storage media (floppy disk or EPROM) to main memory. After the transfer is complete, CPBOOT will reside in the first 32 kilowords of main memory. Execution will begin at location

'1000. Therefore, the hardware required for CPBOOT includes a functional main memory, and the media on which CPBOOT resides. In addition, CPBOOT requires a central processor capable of program execution in 64V-mode, supporting at least the hard core instructions as defined in PE-T-1055. The hard core instructions are used to report the test results of a hardware failure. The diagnostic processor must be capable of communication with the system console. A functional I/O device is needed for the system console. The diagnostic processor makes no assumptions about system console characteristics; the system console may be either hardcopy or CRT. Also, baud rate requirements have been made transparent by the diagnostic processor. The system console device address is assumed to be "04.

4.2 Required for Successful Completion of CPBOOT

The selected boot device must be available for use; the supporting controller must be present in the system; and interconnecting cables must be attached. Also, the boot device must have the appropriate media mounted, and the boot device must be set on-line and ready. In the special case of the PAL boot, and I/O bus tester, a host system, and the interconnecting cable are required.

4.3 Required for Display of Failure Report

Because no testing can be performed on the diagnostic processor, via PMA, CPBOOT assumes coherence between the diagnostic processor and the system console.

4.4 Compatibility

For compatibility, CPBOOT continues to provide the same method for booting from storage module disk, magnetic tape, and Prime Automatic Load (PAL), as the early versions of this program. CPBOOT continues to use a minimum of controller hardware required to perform the boot sequence. Thereby, any generic type, 4004/4005 disk controller, magnetic tape controller, I/O Bus Tester, or diagnostic processor, will be supported by CPBOOT. See Appendix A.

Disk controllers must have the device addresses '26, '27, '22, and '23. The magnetic tape controllers must have the device addresses '14 and '13.

CPBOOT will not support Option A, Option B', floppy disk, fixed head disk, paper tape, or 7-track magnetic tape, as these devices are not supported by PRIMOS.

5 Software Interface

CPBOOT will determine the destination for each type of media transfer.

The boot program loaded from moving head disk will be transferred to main memory starting at location '000760. Because the disk record size used by PRIMOS is 1040 words, a total of 1040 words (one record) will be transferred via DMA. The first 16 words transferred consist of the "disk record header", thus the starting address of the transfer is location '000760. Upon completion of the transfer, execution of the boot program (from disk) starts at location '1000.

The boot program loaded from magnetic tape will be transferred to main memory starting at location '000200. The first record from magnetic tape will be transferred via DMA. Upon completion of the transfer, execution of the boot program (from tape) starts at location '1000.

The boot program loaded from the "I/O Bus Tester" will be transferred to main memory starting at location "017000. A total of 512 words will be transferred via PIO. Upon completion of the transfer, execution of the boot program (from the I/O Bus Tester) starts at location "017000.

All device specific code (I/O drivers) is self contained within CPBOOT. Prior to execution of the new boot program overlay, registers are not initialized with "RVEC" data and execution is continued in V-mode.

6 Design Constraints

CPBOOT will occupy no more than 32 kilowords of storage.

In the event of a failure, CPBOOT will provide the user with an error message indicating what was being tested, the expected results, and the actual results. CPBOOT will provide, as an option, the ability to enable an informative message display. For more detail about the informative message display, see the section on "Display Formats".

CPBOOT will not identify FRU (Field Replaceable Unit) components. CPBOOT does not provide "scope loop" capability.

CPBOOT will provide the user with diagnostic coverage for V-mode addressing, and will provide, as an option, the ability to bypass diagnostic testing. See Appendix B.

7 Assumptions

The Memory Test assumes the memory size to be at least twice the cache size. MTBOOT produced by MAGSAV and used for magnetic tape boot is assumed to be built by PRIMOS REV 20.

8 Differences

The visual appearance of CPBOOT to the user remains the same as the earlier version of this program, with two exceptions. Upon a hardware failure, an error message will be displayed at the system console, rather than an unfriendly halt with no error message. Also, the user has the option of displaying informative messages while CPBOOT executes.

CPBOOT provides additional testing, prior to the boot procedure, to verify the integrity of the hardware. During the boot procedure, additional testing continues to provide a constant check of hardware conditions.

CPBOOT assumes a new sense switch definition to include two additional disk controllers. See Appendix A for "Storage Module '22" and "Storage Module '23". The sense switch (11) used here for previously a "don't care".

9 User Interface

CPBOOT is automatically invoked upon power up. CPBOOT may also be invoked upon power up. CPBOOT may also be invoked manually, via the diagnostic processor command, "BOOT". This command supports the boot device arguments. See Appendix A.

To lessen the impact on the user, the silent appearance of CPBOOT will remain the same as the earlier version, with the two exceptions. In the first exception, any subsequent hardware failures during the boot process will be reported at the system console. The intention here, is to assist the user in diagnosing the problem. In the second exception, the user may enable the CPBOOT option to display informative messages. The informative messages are intended to display the progress of CPBOOT as execution proceeds. The option is enabled via "data switch" setting. The diagnostic processor command line is: BOOT [arg1] [arg2], where [arg1] is the "sense switch" value, and [arg2] is the "data switch" value.

When the central processor is unable to perform a rudimentary task, such as successful execution of the hard core instructions required for communication to the system console, then there is no alternative but to proceed with an unfriendly halt. Also, multiple checks, traps, and faults will be terminated with a halt to avoid the potential hazard of looping indefinitely within the handler routine, after attempting to display the Failure Report.

A warning message will appear at the system console when CPBOOT detects a microverify failure on any of the I/O controllers. This warning may not be fatal, provided the controller is not needed to complete the boot sequence.

The only change visible to the user, therefore, is the reporting of a hardware failure rather than halting on a hardware failure, and the optional display of informative messages.

All messages are in English.

10 Display Formats

10.1 Informative Messages

By default, the informative messages are disabled (suppressed). The user may request the informative messages to be displayed via a data switch setting. See appendix B. Informative messages will display a short description of each test case as CPBOOT executes. The user may also request a forced display of error messages. This forced display of error messages will display the intermediate results of each test case regardless of the results.

With the informative display turned on, the user is prompted with the first message: "Beginning Control Panel Boot". Subsequent test case messages, such as, "Testing: Register File", "Testing: Memory", or "Testing: Disk Controller '26, Disk Unit 0, channel order SELECT", will be displayed. Upon successful completion of the boot routine, the user is prompted with the final message: "Control Panel Boot Completed". (See the following sample display)

Beginning Control Panel Boot. [Rev 1.0]
Testing: Register File.
Testing: Register File Address.
Testing: Address Trap Mechanism.
Testing: 64V Base Register Relative Indirect.
Testing: 64V Base Register Relative Indirect Preindexed.
Testing: 64V Procedure Relative Indirect Postindexed.
Testing: 64V Two Word Indirect.
Testing: 64V Two Word Indexed by Y.
Testing: 64V Two Word Indirect Preindexed by X.
Testing: 64V Two Word Indirect Postindexed by X.
Testing: Base Registers.
Testing: 64V Two Word Direct using XB.
Testing: Memory
Testing: I/O controller sanity.
Boot Procedure.
Testing: Existence of controller: '000026
Testing: Controller "000026, Unit 000000, SELECT
Testing: OTA ready.
Testing: Controller '000026, Unit 000000, STATUS
Testing: OTA ready.
Testing: Controller '000026, Unit 000000, SEEK
Testing: OTA ready.
Testing: Controller '000026, Unit 000000, STATUS
Testing: OTA ready.
Testing: Controller '000026, Unit 000000, READ
Testing: OTA ready.
Testing: Controller '000026, Unit 000000, STATUS
Testing: OTA ready
Testing: BOOT Record Header.
Testing: BOOT Revision Compatibility
Control Panel Boot Completed.

[Failure Report]
Testing: Existence of controller: '0000xx
Actual: ABSENT
Expected: PRESENT

[Failure Report]
Testing: Microverify. controller '0000xx
Actual: FAIL
Expected: PASS

[Failure Report]
Testing: OTA ready.
Actual: TIME OUT
Expected: READY

[Failure Report]
Testing: Controller '0000xx, Unit 00000x, STATUS
Actual: 100000
Expected: 100000

[Failure Report]
Testing BOOT Record Header
[address '000760 - '000763]
Actual: 000000 000000 000000 000000
Expected: 000000 000000 000000 000001

[Failure Report]
Testing: BOOT Revision Compatibility
Actual: INCOMPATIBLE
Expected: COMPATIBLE

[Failure Report]
Testing: MTBOOT Revision Compatibility
Actual: INCOMPATIBLE
Expected: COMPATIBLE

[Failure Report]
Testing: INA ready
Actual: NOT READY
Expected: READY

[Failure Report]
Testing: Memory
Address: 000000
Actual: 000000
Expected: 000000

Machine Check Flag
Actual: SET
Expected: RESET

DSWSTAT: 000000/000000
DSWPARITY: 000000/000000
DSWRMA: 000000/000000

[Failure Report]
Testing: INA ready
Actual: TIMEOUT
Expected: READY

[Failure Report]
Testing: Sense switch setting.
Must be at least '100000.

[Failure Report]
Testing: Register File.
Address: 000000
Actual: 000000
Expected: 000000

[Failure Report]
Testing: Register File Address.
Address: 000000
Actual: 000000
Expected: 000000

[Failure Report]
Testing: Address Trap Mechanism.
Address: 000000
Actual: 000000
Expected: 000000

[Failure Report]
Testing: 64V Base Register Relative Indirect.
Initial A-register: 000000
Actual: 000000
Expected: 000000

[Failure Report]
Testing: 64V Base Register Relative Indirect Preindexed.
X-register: 000000
Actual: 000000
Expected: 000000

[Failure Report]
Testing: 64V Procedure Relative Indirect Postindexed.
X-register: 000000
Actual: 000000
Expected: 000000

[Failure Report]
Testing: 64V Two Word Indirect.
Indirect Pointer: 000000
Actual: 000000
Expected: 000000

[Failure Report]
Testing: 64V Two Word Indexed by Y.
Y-register: 000000
Actual: 000000
Expected: 000000

[Failure Report]
Testing: 64V Two Word Indirect Preindexed by X.
X-register: 000000
Actual: 000000
Expected: 000000

[Failure Report]
Testing: 64V Two Word Indirect Postindexed by X.
X-Register: 000000
Actual: 000000
Expected: 000000

[Failure Report]
Testing: Base Registers.
Base Register: 000000
Actual: 000000/000000
Expected: 000000/000000

[Failure Report]
Testing: 64V Two Word direct using XB.
XB Register: 000000/000000
Actual: 000000
Expected: 000000

[Failure Report]
MACHINE CHECK
Program Counter: 000000

DSWSTAT: 000000/000000
DSWPARITY: 000000/000000
DSWRMA: 000000/000000

[Failure Report]
MEMORY PARITY
Program Counter: 000000

DSWSTAT: 000000/000000
DSWPARITY: 000000/000000
DSWRMA: 000000/000000

[WARNING] Controller '0000xx has failed microverify.

10.2 Error Messages

The Failure Report messages detail what is being tested, the results expected, and the results actually obtained. The Customer Service Representative is the primary target of the Failure Report. However, manufacturing should also benefit from

the Failure Report. (See the following sample Failure Reports)
Upon reporting the error message, CPBOOT will halt. At this
time, CPBOOT can be continued via the "RUN" command. However,
the results are unpredictable since the hardware is failing. In
cases when the RUN command is not permitted after the Failure
Report, the user will be prompted with the message, "SYSCLR
required.", and the boot sequence must be restarted.

11 Performance

CPBOOT requires approximately 11 seconds to complete execution.
Note that 10 of these seconds are spent waiting for completion of
I/O controller initialization.

12 CPBOOT Algorithm

This is the basic program flow of CPBOOT. Note the diagnostic
testing is performed prior to the boot procedure, and is
continued in the boot procedure.

```
Begin CPBOOT
/*Perform hardware verification*/
Test the hard core instruction subset.
If unsuccessful then HALT.

Test the CPBOOT checksum.
If unsuccessful then HALT.

Clean memory parity.
Enable machine check mode.
Setup the system console.
/*Failure report handler is now enabled. */
If NOT bypass diagnostics
Then
/*If the following diagnostics are unsuccessful*/
/*then a failure report will be displayed. */
/*followed by a HALT. */

Test the register file.
Test v-mode addressing.
Test memory.

Perform sanity test on all controllers.
End
/*Perform the boot procedure*/
End CPBOOT
```

13 Hardware Verification

The hard core instructions are tested to ensure that at least the kernel code and the error message handler of CPBOOT will execute successfully. The checksum test will detect a main memory failure caused by bad ram or a program load failure caused by a faulty diagnostic processor boot strap mechanism. The register file plays an integral part in the various memory references, and therefore, the register file is tested for data and address integrity. Also, the address trap mechanism is tested. Because the disk boot strap program, "BOOT", requires V-mode addressing from the onset, CPBOOT will provide V-mode addressing tests, which include V-mode memory reference, V-mode PIO, and a subset of branch instructions. The memory is checked to verify its basic function of storing and retrieving data. The emphasis, here, is placed on data integrity. A "sanity" test is performed on the new I/O controllers that provide CPBOOT with information regarding the pass/fail condition of the on-board microverify. CPBOOT will inform the user of I/O controller hardware failures that a system crash, due to a bold attempt by PRIMOS to use broken hardware, will be less surprising.

13.1 Hard Core Instructions

The minimum of instructions required for communication to the system console will be tested. These hard core instructions are described in PE-T-1055. If one of these hard core instructions fails, there is no alternative but to halt. This and multiple checks, traps, and faults are the only unfriendly part of CPBOOT.

13.2 Checksum

A checksum of CPBOOT will be done to verify a successful bootstrap transfer by the diagnostic processor. Bad memory cells, within the program area of CPBOOT, will also be detected here. Any subsequent failure will cause a halt.

13.3 System console

Upon successful completion of the hardcore instruction testing, and checksum test, the PIO instruction, "INA" will be attempted to establish the existence of I/O device 4. further, the diagnostic processor is initialized for half-duplex. From this point, error messages can now be displayed to the user.

13.4 Register File Test

This test consists of a write to, then read from, each register file location of the current register set. The data patterns consist of floating ones and floating zeroes. Also, a register file unique address test will be done. Finally, the address trap mechanism will be checked, to ensure that the register file is being used rather than memory.

13.5 V-Mode Addressing Test

This test will verify 64V Procedure Relative direct, indirect, indexed, and indirect postindexed addressing modes. The 64V Base Register Relative mode will include direct and indirect address testing. Also, 64V Two Word Memory Reference direct, indirect indexed by X, indexed by Y, preindexed by X, and postindexed by X will be tested. The "EIO" instruction (execute I/O) will be tested using the branch instructions, "BCNE" and "BCEQ", to check the condition codes.

13.6 Memory Test

The first 64 kilowords of memory will be tested, using a data pattern of alternating ones and zeroes. Because this is a non-destructive test, the initial data at each test location will be saved prior to testing, then restored after testing.

After using a temporary location to save the initial data from the test location, the first test data pattern is written to the test location. To ensure a read from memory, rather than from cache, the same cache locations is invalidated (flushed), then the read is performed on the test memory location. Next, the resultant data from the memory read operation is compared to the test data pattern. The initial data, in the temporary location, is then restored to the test location. Because machine check mode is not enabled here, no memory parity traps can occur. However, the "machine check flag" is checked, as an indication of a memory parity error. This process is repeated using the complement data pattern. The test address range is from '000000 to '177777.

On the assumption that the memory size is at least twice the size of cache, the cache invalidation routine used in the memory test will support any cache size.

13.7 Sanity Test

Each controller (addressed '01 to '76) will be polled and checked for a ready responses. If the controller responds ready and the controller has the new 32-bit "ID" longword, then the "LED" bits, from the resultant "ID" word, are checked for the condition: "divide is functioning correctly". If the "ID" word indicates something other than this condition, a warning message is displayed at the system console, indicating that the controller has failed microverify. At this point, the error is not considered critical, and CPBOOT continues.

14 Boot Procedure

First, the boot device, indicated by sense switches [14-16] (APPENDIX A), will be checked. If the selected boot device is not supported, the user will be informed at the system console. CPBOOT then halts execution, such that the user may select a boot

device that is supported by CPBOOT. The user must reenter the "BOOT" command and any arguments.

If the selected boot device is supported, the supporting controller for the selected boot device, will be polled. If the poll fails the ready condition, then the user will be informed that the controller is absent from the system. next if the controller has the new 32-bit "ID" longword, then the "LED" bits, from the resultant "ID" word, are checked for the condition: "device is functioning correctly". If the "ID" word indicates something other than this condition, a warning message is displayed at the system console, indicating that the controller has failed microverify. At this point, the error is considered critical, and CPBOOT halts. If the "ID" word is okay, then singular controller functions will be attempted, as part of the boot procedure, and as part of the hardware verification. Completion of each function is expected. In order to avoid the potential infinite loop, PIO ready conditions are expected to respond within a given period of time. The time delay is cpu model sensitive. A data switch completion of the controller function, a status check will be taken from the controller. Unsuccessful completion of a controller function or a status error as a result of the attempted controller function will be reported to the user. Upon successful completion of the boot procedure, execution continues with the boot program overlay, obtained from the boot device media. From this point, the CPBOOT function is completed.

No attempt is made to turn on or turn off burst mode on the controller. No retry is attempted after a read failure. However, a retry can be accomplished by rebooting.

15 Appendix A

Sense Switches						Definition		
0	000	000	001	111	111			
1	234	567	890	123	456			
.	000	Start at address in sense switches [01-10]		
.	001	Not supported		
.	010	Not supported		
.	011	Not supported		
.	001	001	100	Storage Module '26	Unit 0	
.	001	001	100	Storage Module '26	Unit 1	
.	101	001	100	Storage Module '26	Unit 2	
.	111	001	100	Storage Module '26	Unit 3	
.	001	011	100	Storage Module '27	Unit 0	
.	011	011	100	Storage Module '27	Unit 1	
.	101	011	100	Storage Module '27	Unit 2	
.	111	011	100	Storage Module '27	Unit 3	
.	001	101	100	Storage Module '22	Unit 0	
.	011	101	100	Storage Module '22	Unit 1	
.	101	101	100	Storage Module '22	Unit 2	
.	111	101	100	Storage Module '22	Unit 3	
.	001	111	100	Storage Module '23	Unit 0	
.	011	111	100	Storage Module '23	Unit 1	
.	101	111	100	Storage Module '23	Unit 2	
.	111	111	100	Storage Module '23	Unit 3	
.	00.	.0.	101	Magnetic Tape '14	Unit 0	9 track
.	01.	.0.	101	Magnetic Tape '14	Unit 1	9 track
.	10.	.0.	101	Magnetic Tape '14	Unit 2	9 track
.	11.	.0.	101	Magnetic Tape '14	Unit 3	9 track
.	00.	.1.	101	Magnetic Tape '14	Unit 0	9 track
.	01.	.1.	101	Magnetic Tape '13	Unit 1	9 track
.	10.	.1.	101	Magnetic Tape '13	Unit 2	9 track
.	11.	.1.	101	Magnetic Tape '13	Unit 3	9 track
.	111	Pal Boot (in-house only)		
.1	Do not enter machine check mode		

Each "." is a "don't care".

