

NAME

70boot – 11/70 bootstrap procedures

DESCRIPTION

To bootstrap programs from a wide range of storage media, the PDP-11/70 has a dedicated diagnostic bootstrap loader called the M9301-YC. The M9301-YC contains two 256 word ROMs (17 765 000 to 17 765 776 and 17 773 000 to 17 773 776) which contain hardware verification diagnostic routines and bootstrap loader routines.

The diagnostic portion tests the basic CPU to verify correct operation. The branches, registers, all addressing modes, and most of the instructions are checked. If requested, memory management and the UNIBUS map are turned on. Then memory is tested from virtual address 001 000 to 157 776 with the cache disabled. Next the cache is enabled and tested.

The physical memory tested is determined by the console switches. Console switches <15:12> are used to set physical address bits <19:16>. If console switches <15:12> are zero, memory management and the UNIBUS map will not be enabled, so that physical memory 0 to 157 776 will be used. If console switches <15:12> are non-zero, then memory management, the UNIBUS map, and 22-bit mapping will be enabled. Table I describes the physical address ranges for each switch setting. In all cases, virtual addresses 160 000 to 177 776 are mapped to the peripheral page, physical addresses 17 600 000 to 17 777 776. Note that physical memory above 512K words is not accessible by this program even though the physical memory maximum is 1920K words.

The bootstrap portion of the M9301-YC attempts to BOOT from the device and drive number specified in the console switches. Console switches <7:3> select the device and console switches <2:0> select the drive number. Table II describes the devices selected for each switch setting. If console switches <7:0> are zero, the program will read a set of switches on the M9301-YC, set by field service, to determine a default boot device and drive number. These switches appear at location 17 773 024, however bits <8:4> select the device and bits <3:1> select the drive number.

Having selected a boot device, the program will read a block of data into memory starting at virtual address 0, and then jump to virtual address 0. Table III describes the details of booting for each device. Note that the physical address selection is the same as described above for the diagnostic portion. Excluding the RX11/RX01 floppy disk, bootstrap programs must fit in one block of 256 words, even though this program may read in more.

To start operation of the bootstrap loader, halt the CPU by depressing the HALT switch, set the Address Display select switch to Console Physical, set the Console Switch Register to 165 000, and depress the Load Address switch. Then reset the console switches to 0 and set switches <15:12> for the desired physical memory (normally 0) and switches <7:0> for the desired device (normally 0 for the default boot). Put the HALT switch in the ENABLE position and depress the START switch. The diagnostic portion will then run followed by the boot from the selected media. This takes approximately three seconds.

Any error during the diagnostic portion will cause the CPU to halt. Table IV lists the addresses and error indications. Only cache errors are recoverable in that by pressing the CONTINUE switch the program will disable the cache by forcing misses and proceed to the bootstrap section. If there is an error in reading the boot block, the program will do a RESET instruction and jump back to the memory test section (test 24) and then attempt to boot again.

SEE ALSO

unixboot(VIII)

Table I – Physical Memory Selection

Console switches <15:12>	Physical addresses
00	00 000 000 - 00 157 776
01	00 200 000 - 00 357 776
02	00 400 000 - 00 557 776
03	00 600 000 - 00 757 776
04	01 000 000 - 01 157 776
05	01 200 000 - 01 357 776
06	01 400 000 - 01 557 776
07	01 600 000 - 01 757 776
10	02 000 000 - 02 157 776
11	02 200 000 - 02 357 776
12	02 400 000 - 02 557 776
13	02 600 000 - 02 757 776
14	03 000 000 - 03 157 776
15	03 200 000 - 03 357 776
16	03 400 000 - 03 557 776
17	03 600 000 - 03 757 776

Table II – Device selection

Console switches <7:3>	Device
00	illegal
01	TM11/TU10 Magnetic tape
02	TC11/TU56 DECtape
03	RK11/RK05 Disk pack
04	RP11/RP03 Disk pack
05	reserved
06	RH70/TU16 Magnetic tape
07	RH70/RP04 Disk pack
10	RH70/RS04 Fixed head disk
11	RX11/RX01 Diskette
12-37	illegal

Table III – Boot procedures

TU10:	Select drive, wait until online, set to 800 bpi, rewind, space forward 1 record, read 1 record (maximum of 256 words).
TU56:	Select drive, rewind, read 512 words.
RK05 or	
RP03:	Select drive, start at block 0, read 512 words.
TU16:	Select drive on first TM02, wait until online, set to 800 bpi, PDP format, rewind, space forward 1 record, read 1 record (maximum of 512 words).
RP04:	Select drive, read-in preset, set to 16-bits/word, ECC inhibit, start at block 0, read 512 words.
RS04:	Select drive, start at block 0, read 512 words.
RX01:	Select drive 0 or 1, start at track 1, sector 1 (IBM standard), read 64 words.

Table IV – Error halts

Address displayed	Test	Subsystem under test
17 765 004	1	Branch
17 765 020	2	Branch
17 765 036	3	Branch
17 765 052	4	Branch
17 765 066	5	Branch
17 765 076	6	Branch
17 765 134	7	Register data path
17 765 146	10	Branch
17 765 166	11	CPU instruction
17 765 204	12	CPU instruction
17 765 214	13	CPU instruction
17 765 222	14	CPU instruction
17 765 236	14	CPU instruction
17 765 260	15	CPU instruction
17 765 270	16	Branch
17 765 312	16	CPU instruction
17 765 346	17	CPU instruction
17 765 360	20	CPU instruction
17 765 374	20	CPU instruction
17 765 450	21	Kernel PAR
17 765 474	22	Kernel PDR
17 765 510	23	JSR
17 765 520	23	JSR
17 765 530	23	RTS
17 765 542	23	RTI
17 765 550	23	JMP
17 765 742	25	Main memory data compare error
17 765 760	25	Main memory data compare error
17 776 000	25	Main memory parity error; no recovery possible from this error
17 773 644	26	Cache memory data compare error
17 773 654	26	Cache memory no hit, recoverable
17 773 736	27	Cache memory data compare error
17 773 746	27	Cache memory no hit, recoverable
17 773 764	25/26	Cache memory parity error, recoverable