DM&P M6117D BIOS Post Codes

2003-10-30

Following is the checkpoint list in AMIBIOS in order of execution.

Uncompressed INIT code checkpoints

Check Point	Description
D0	NMI is Disabled. CPU ID saved. Init code Checksum verification starting.
D1	To do DMA init, Keyboard controller BAT test, start memory refresh and going to 4GB flat mode.
D3	To start Memory sizing.
D4	To comeback to real mode. Execute OEM patch. Set stack.
D5	E000 ROM enabled. Init code is copied to segment 0 and control to be transferred to segment 0.
D6	Control is in segment 0. To check <CTRL><HOME> key and verify main BIOS checksum. If either <CTRL><HOME> is pressed or main BIOS checksum is bad, go to check point E0 else go to check point D7.
D7	Main BIOS runtime code is to be decompressed and control to be passed to main BIOS in shadow RAM.

Boot Block Recovery Code Check Points

Check Point	Description
E0	Onboard Floppy Controller (if any) is initialized. To start base 512K memory test.
E1	To initialize interrupt vector table.
E2	To initialize DMA and interrupt controllers.
E6	To enable floppy and timer IRQ, enable internal cache.
ED	Initialize floppy drive.
EE	Start looking for a diskette in drive A: and read 1st sector of the diskette.
EF	Floppy read error.
F0	Start searching 'AMIBOOT.ROM' file in root directory.
F1	'AMIBOOT.ROM' file not present in root directory.
F2	Start reading FAT table and analyze FAT to find the clusters occupied by 'AMIBOOT.ROM' file.
F3	Start reading 'AMIBOOT.ROM' file cluster by cluster.
F4	'AMIBOOT.ROM' file not of proper size.
F5	Disable internal cache.
FB	Detect Flash type present.
FC	Erase Flash.
FD	Program Flash.
FF	Flash program successful. BIOS is going to restart.

Runtime code is uncompressed in F000 shadow ram

Check Point	Description
03	NMI is Disabled. To check soft reset/power-on.
05	BIOS stack set. Going to disable Cache if any.
06	POST code to be uncompressed.
07	CPU init and CPU data area init to be done.
08	CMOS checksum calculation to be done next.
0B	Any initialization before keyboard BAT to be done next.
0C	KB controller I/B free. To issue the BAT command to keyboard controller.
0E	Any initialization after KB controller BAT to be done next.
0F	Keyboard command byte to be written.
10	Going to issue Pin-23, 24 blocking/unblocking commands.
11	Going to check pressing of, key during power-on.
12	To init CMOS if "Init CMOS in every boot" is set or key is pressed. Going to disable DMA and Interrupt controllers.
13	Video display is disabled and port-B is initialized. Chipset init about to begin.
14	8254 timer test about to start.
19	About to start memory refresh test.
1A	Memory Refresh line is toggling. Going to check 15us ON/OFF time.
23	To read 8042 input port and disable Megakey GreenPC feature. Make BIOS code segment writeable.
24	To do any setup before Int vector init.
25	Interrupt vector initialization about to begin. To clear password if necessary.
27	Any initialization before setting video mode to be done.
28	Going for monochrome mode and color mode setting.
2A	Different BUSes init (system, static, output devices) to start if present. (Please see Appendix for details of different BUSes).
2B	To give control for any setup required before optional video ROM check.
2C	To look for optional video ROM and give control.
2D	To give control to do any processing after video ROM returns control.
2E	If EGA/VGA not found then do display memory R/W test.
2F	EGA/VGA not found. Display memory R/W test about to begin.
30	Display memory R/W test passed. About to look for the retrace checking.
31	Display memory R/W test or retrace checking failed. To do alternate Display memory R/W test.
32	Alternate Display memory R/W test passed. To look for the alternate display retrace checking.
34	Video display checking over. Display mode to be set next.
37	Display mode set. Going to display the power on message.
38	Different BUSes init (input, IPL, general devices) to start if present. (Please see Appendix for details of different BUSes).

39	Display different BUSes initialization error messages. (Please see Appendix for details of different BUSes).
3A	New cursor position read and saved. To display the Hit
40	To prepare the descriptor tables.
42	To enter in virtual mode for memory test.
43	To enable interrupts for diagnostics mode.
44	To initialize data to check memory wrap around at 0:0.
45	Data initialized. Going to check for memory wrap around at 0:0 and finding the total system memory size.
46	Memory wrap around test done. Memory size calculation over. About to go for writing patterns to test memory.
47	Pattern to be tested written in extended memory. Going to write patterns in base 640k memory.
48	Patterns written in base memory. Going to find out amount of memory below 1M memory.
49	Amount of memory below 1M found and verified. Going to find out amount of memory above 1M memory.
4B	Amount of memory above 1M found and verified. Check for soft reset and going to clear memory below 1M for soft reset. (If power on, go to check point# 4Eh).
4C	Memory below 1M cleared. (SOFT RESET) Going to clear memory above 1M.
4D	Memory above 1M cleared. (SOFT RESET) Going to save the memory size. (Go to check point# 52h).
4E	Memory test started. (NOT SOFT RESET) About to display the first 64k memory size.
4F	Memory size display started. This will be updated during memory test. Going for sequential and random memory test.
50	Memory testing/initialization below 1M complete. Going to adjust displayed memory size for relocation/ shadow.
51	Memory size display adjusted due to relocation/ shadow. Memory test above 1M to follow.
52	Memory testing/initialization above 1M complete. Going to save memory size information.
53	Memory size information is saved. CPU registers are saved. Going to enter in real mode.
54	Shutdown successful, CPU in real mode. Going to disable gate A20 line and disable parity/NMI.
57	A20 address line, parity/NMI disable successful. Going to adjust memory size depending on relocation/shadow.
58	Memory size adjusted for relocation/shadow. Going to clear Hit
59	Hit
60	DMA page register test passed. To do DMA#1 base register test.
62	DMA#1 base register test passed. To do DMA#2 base register test.
65	DMA#2 base register test passed. To program DMA unit 1 and 2.
66	DMA unit 1 and 2 programming over. To initialize 8259 interrupt controller.
7F	Extended NMI sources enabling is in progress.
80	Keyboard test started. clearing output buffer, checking for stuck key, to issue keyboard reset command.

81	Keyboard reset error/stuck key found. To issue keyboard controller interface test command.
82	Keyboard controller interface test over. To write command byte and init circular buffer.
83	Command byte written, Global data init done. To check for lock-key.
84	Lock-key checking over. To check for memory size mismatch with CMOS.
85	Memory size check done. To display soft error and check for password or bypass setup.
86	Password checked. About to do programming before setup.
87	Programming before setup complete. To uncompress SETUP code and execute CMOS setup.
88	Returned from CMOS setup program and screen is cleared. About to do programming after setup.
89	Programming after setup complete. Going to display power on screen message.
8B	First screen message displayed. Message displayed. PS/2 Mouse checks and extended BIOS data area allocation to be done.
8C	Setup options programming after CMOS setup about to start.
8D	Going for hard disk controller reset.
8F	Hard disk controller reset done. Floppy setup to be done next.
91	Floppy setup complete. Hard disk setup to be done next.
95	Init of different BUSes optional ROMs from C800 to start. (Please see Appendix-I for details of different BUSes).
96	Going to do any init before C800 optional ROM control.
97	Any init before C800 optional ROM control is over. Optional ROM check and control will be done next.
98	Optional ROM control is done. About to give control to do any required processing after optional ROM returns control and enable external cache.
99	Any initialization required after optional ROM test over. Going to setup timer data area and printer base address.
9A	Return after setting timer and printer base address. Going to set the RS-232 base address.
9B	Returned after RS-232 base address. Going to do any initialization before Coprocessor test.
9C	Required initialization before Coprocessor is over. Going to initialize the Coprocessor next.
9D	Coprocessor initialized. Going to do any initialization after Coprocessor test.
9E	Initialization after Coprocessor test is complete. Going to check extd keyboard, keyboard ID and num-lock. Keyboard ID command to be
A2	Going to display any soft errors.
A3	Soft error display complete. Going to set keyboard type mantic rate.
A4	Keyboard type mantic rate set. To program memory wait states.
A5	Going to enable parity/NMI.
A7	NMI and parity enabled. Going to do any initialization required before giving control to optional ROM at E000.
A8	Initialization before E000 ROM control over. E000 ROM to get control next.
A9	Returned from E000 ROM control. Going to do any initialization required after E000 optional ROM control.
AA	Initialization after E000 optional ROM control is over. Going to display the system configuration.

AB	To uncompress DMI data and execute DMI POST init.
B0	System configuration is displayed.
B1	Going to copy any code to specific area.
00	Copying of code to specific area done. Going to give control to INT-19 boot loader.

The system BIOS gives control to the different BUSes at following checkpoints to do various tasks on the different BUSes.

CHECK-POINT	DESCRIPTION OF CHECK-POINT
2A	Different BUSes init (system, static, output devices) to start if present.
38	Different BUSes init (input, IPL, general devices) to start if present.
39	Display different BUSes initialization error messages.
95	Init of different BUSes optional ROMs from C800 to start.

While control is inside the different BUS routines, additional checkpoints are output to port 80h as WORD to identify the routines under execution. These are WORD checkpoints, the LOW BYTE of checkpoint is the system BIOS checkpoint from where the control is passed to the different BUS routines and the HIGH BYTE of checkpoint is the indication of which routine is being executed in different BUSes. The details of HIGH BYTE of these checkpoints are as follows:

HIGH BYTE XY

the upper nibble 'X' indicates the function# is being executed. 'X' can be from 0 to 7.

```
0 = func#0, disable all devices on the BUS concerned.
1 = func#1, static devices init on the BUS concerned.
2 = func#2, output device init on the BUS concerned.
3 = func#3, input  device init on the BUS concerned.
4 = func#4, IPL    device init on the BUS concerned.
5 = func#5, general device init on the BUS concerned.
6 = func#6, error reporting for the BUS concerned.
7 = func#7, add-on ROM init for all BUSes.
```

the lower nibble 'Y' indicates the BUS on which the different routines are being executed. 'Y' can be from 0 to 5.

```
0 = Generic DIM (Device Initialization Manager).
1 = On-board System devices.
2 = ISA devices.
3 = EISA devices.
4 = ISA PnP devices.
5 = PCI devices.
```

AMIBIOS 071595 [Enhanced] Check Point List

Technical Support

For more technical support, please visit http://www.dmp.com.tw/tech or mail to tech@dmp.com.tw.