QuickPCI POST Diagnostic Card

USER MANUAL

Table of contents:

Introduction

QuickPCI POST Diagnostic Card

With the QuickPCI POST Card, you now have an efficient, high quality and easy to operate diagnostic card at your disposal. This was designed for the PCI bus system, which has become standard and even enables you to diagnose "dead" PC's quickly. "Dead PC's" are no longer a problem with the QuickPCI POST Card, and neither are those time consuming intermittent power supply problems. Do you want to know more about the ATX timing or the PCI? Simply get plugged in. QuickPCI POST Card also comes with a 6 months manufacture warranty.

PCI Interface

As for the interface of the PCI bus, the QuickPCI POST Card was designed with a fast and programmable PLD-chip, which takes over decoding and monitoring of various bus signals. A special consideration during development was not only high functionality but also the future reliability of the design.

Power On Self Test (POST) Display

QuickPCI POST Card features a 2-digit-hexadecimal-display. The extra bright display can even be read easily in dim light and features the monitoring of I/O Port address 80h, which is used in sequence from BIOS of PC booting to out put of POST codes. If the BIOS failed to boot, possibly due to a hard ware

problem, the corresponding POST error code will give you information about the cause of the problem.

Bus Tension Control

To control the PCI bus tension, QuickPCI POST Card offers a variety of ways. Four LED's show the existing voltage +5V, +12V, -12V and 3.3 Volt.

Reset and Clock Signal Monitoring

To monitor the bus signal and reset line (PCI clock) the clock signal is under continuous monitoring of its two possible states ("0" logical low and "1" logical high). Simpler diagnostic boards often only check the first cycle of the clock and reset signal. QuickPCI POST Card shows the actual states of clock and reset signals using LED's. By using the PCI reset signal, any PC hardware device (chipset, cpu, graphics controller, and etc.) is reset to a defined state. A hardware failure on the main board or an add on card can cause the reset signal to stick to its active state. This event is called "Reset Loop" and prevents the PC system from booting. Reset Loops can be detected by watching the reset LED status.

Feature Overview

- Jumper less and easy to operate design
- Real time monitoring of clock and reset signal
- LED display for +12v, -12v, 3.3v and 5v bus voltages
- Gold overlay of all mechanical contacts
- 6 months warranty

AWARD Elite (Version 4.51PG):

01	Processor test; Processor status verification
02	Processor test 2; Read/Write and verify all CPU registers
03	Initialize chips; Disable NMI, PIE, AIE, UEI, SQWV. Disable video, parity checking, DMA. Reset math coprocessor. Clear all page registers and CMOS shutdown. Initialize DMA controller 0 and 1. Initialize interrupt controllers 0 and 1.
04	Test memory refresh toggle
05	Blank video, initialize keyboard; Keyboard controller initialization
07	Test CMOS interface and battery
08	Set up low memory; Early chipset initialization, memory presence test, OEM chipset routines, clear low 64K memory, test first 64K memory
09	Early cache initialization; Cyrix CPU specific, CPU and cache initialization
0A	Set up interrupt vector table; Initialize first 120 interrupt vectors
0B	Test CMOS RAM checksum
0C	Initialize keyboard; Detect the type of keyboard controller
0D	Initialize video interface; Detect CPU clock, read CMOS location 14h to find the type of video in use, detect and initialize video adapter
0E	Test video memory; Write sign-on message to screen, setup shadow RAM
0F	Test DMA controller 0; BIOS checksum test, keyboard detect and initialization
10	Test DMA controller 1
11	Test DMA page registers
12-13	Reserved
14	Test timer counter 2
15	Test 8259-1 mask bits
16	Test 8259-2 mask bits
17	Test stuck 8259 interrupt bits; Test stuck key

18	Test 8259 interrupt functionality
19	Test stuck NMI bits (parity I/O check)
1A	Benchmark; Display CPU clock
1B-1E	Reserved
1F	Set EISA mode; If the EISA memory checksum is good then EISA is initialized. If it's not good then ISA tests and clear EISA mode flag
20	Enable slot 0; System board
21-2F	Enable slots 1-15
30	Size base and extended memory; Size the base memory from 256K to 640K and the extended memory above 1MB
31	Test base and extended memory; Test the base memory from 256K to 640K and the extended memory above 1MB using various bit patterns
32	Test EISA extended memory
33-3B	Reserved
3C	Setup enabled
3D	Initialize and install mouse if present
3E	Setup cache controller
40	Display virus protect disable or enable
41	Initialize floppy
42	Initialize hard drive
43	Detect & Init. serial & parallel ports
44	Reserved
45	Detect and Init. math coprocessor
46	Reserved
47	Reserved
48-4D	Reserved
4E	Mfg. POST loop, or display messages
4F	Security password
50	Write CMOS; Write CMOS back to RAM and clear screen
51	Pre-boot enable; Enable parity checking, enable NMI, enable cache

	before boot
52	Initialize option ROM's; Initialize and ROM's present at locations C800h to EFFFFh
53	Initialize time value
60	Setup virus protect
61	Set boot speed
62	Setup numlock
63	Boot attempt
B0	Spurious
B1	Unclaimed NMI
BE	Chipset default initialization; Program chipset registers and power-on BIOS defaults.
BF	Chipset initialization; Reserved
C0	Turn off chipset cache
C1	Memory presence test; OEM specific, test the size of on-board memory
C5	Early shadow; OEM specific, early shadow enable for fast boot
C6	Cache presence test; External cache-size detection test
E1-EF	Setup pages
FF	Boot loader

AWARD Version 6.0 (i810):

CFh	Test CMOS read/write functionality
C0h	Early chipset initialization: Disable shadow RAM, L2 cache (socket 7 and below), program basic chipset registers
C1h	Detect memory: Auto detection of DRAM size, type and ECC, auto detection of L2 cache (socket 7 and below)
C3h	Expand compressed BIOS code to DRAM
C5h	Call chipset hook to copy BIOS back to E000 & F000 shadow RAM
01h	Expand the Xgroup codes located in physical memory address 1000:0
02h	Reserved
03h	Initial Superio_Early_Init switch
04h	Reserved
05h	Blank out screen; Clear CMOS error flag
06h	Reserved
07h	Clear 8042 interface; Initialize 8042 self test
08h	Test special keyboard controller for Winbond 977 series Super I/O chips; Enable keyboard interface
09h	Reserved
0Ah	Disable PS/2 mouse interface (optional); Auto detect ports for keyboard & mouse followed by a port & interface swap (optional); Reset keyboard for Winbond 977 series Super I/O chips
0Bh	Reserved
0Ch	Reserved
0Dh	Reserved
0Eh	Test F000h segment shadow to see whether it is read/write capable or not. If test fails, keep beeping the speaker
0Fh	Reserved
10h	Auto detect flash type to load appropriate flash read/write codes into the run time area in F000 for ESCD & DMI support
11h	Reserved
12h	Use walking 1's algorithm to check out interface in CMOS circuitry.

	Also set real time clock power status and then check for override
13h	Reserved
14h	Program chipset default values into chipset. Chipset default values are MODBINable by OEM customers
15h	Reserved
16h	Initial Early_Init_Onboard_Generator switch
17h	Reserved
18h	Detect CPU information including brand, SMI type (Cyrix or Intel) and CPU level (586 or 686)
19h	Reserved
1Ah	Reserved
1Bh	Initial interrupts vector table. If no special specified, all H/W interrupts are directed to SPURIOUS_INT_HDLR & S/W interrupts to SPURIOUS_soft_HDLR
1Ch	Reserved
1Dh	Initial EARLY_PM_INIT switch
1Eh	Reserved
1Fh	Load keyboard matrix (notebook platform)
20h	Reserved
21h	HPM initialization (notebook platform)
22h	Reserved
23h	Check validity of RTC value; Load CMOS settings into BIOS stack. If CMOS checksum fails, use default value instead; Prepare BIOS resource map for PCI & PnP use. If ESCD is valid, take into consideration of the ESCD's legacy information; Onboard clock generator initialization. Disable respective clock resource to empty PCI & DIMM slots; Early PCI initialization - Enumerate PCI bus number, assign memory & I/O resource, search for a valid VGA device & VGA BIOS, and put it into C000:0
24h	Reserved
25h	Reserved
26h	Reserved

27h	Initialize INT 09 buffer
28h	Reserved
29h	Program CPU internal MTRR (P6 & PII) for 0-640K memory address; Initialize the APIC for Pentium class CPU; Program early chipset according to CMOS setup; Measure CPU speed; Invoke video BIOS
2Ah	Reserved
2Bh	Reserved
2Ch	Reserved
2Dh	Initialize Multilanguage; Put information on screen display, including Award title, CPU type, CPU speed, etc...
2Eh	Reserved
2Fh	Reserved
30h	Reserved
31h	Reserved
32h	Reserved
33h	Reset keyboard except Winbond 977 series Super I/O chips
34h	Reserved
35h	Reserved
36h	Reserved
37h	Reserved
38h	Reserved
39h	Reserved
3Ah	Reserved
3Bh	Reserved
3Ch	Test 8254
3Dh	Reserved
3Eh	Test 8259 interrupt mask bits for channel 1
3Fh	Reserved
40h	Test 9259 interrupt mask bits for channel 2
41h	Reserved
42h	Reserved
43h	Test 8259 functionality

44h	Reserved
45h	Reserved
46h	Reserved
47h	Initialize EISA slot
48h	Reserved
49h	Calculate total memory by testing the last double last word of each 64K page; Program writes allocation for AMD K5 CPU
4Ah	Reserved
4Bh	Reserved
4Ch	Reserved
4Dh	Reserved
4Eh	Program MTRR of M1 CPU; initialize L2 cache for P6 class CPU & program cacheable range; Initialize the APIC for P6 class CPU; On MP platform, adjust the cacheable range to smaller one in case the cacheable ranges between each CPU are not identical
4Fh	reserved
50h	Initialize USB
51h	Reserved
52h	Test all memory (clear all extended memory to 0)
53h	Reserved
54h	Reserved
55h	Display number of processors (multi-processor platform)
56h	Reserved
57h	Display PnP logo; Early ISA PnP initialization and assign CSN to every ISA PnP device
58h	Reserved
59h	Initialize the combined Trend Anti-Virus code
5Ah	Reserved
5Bh	Show message for entering AWDFLASH.EXE from FDD (optional feature)
5Ch	Reserved
5Dh	Initialize Init_Onboard_Super_IO switch; Initialize

	Init_Onboard_AUDIO switch
5Eh	Reserved
5Fh	Reserved
60h	Okay to enter Setup utility
61h	Reserved
62h	Reserved
63h	Reserved
64h	Reserved
65h	Initialize PS/2 mouse
66h	Reserved
67h	Prepare memory size information for function call: INT 15h ax=E820h
68h	Reserved
69h	Turn on L2 cache
6Ah	Reserved
6Bh	Program chipset registers according to items described in Setup & Auto-Configuration table
6Ch	Reserved
6Dh	Assign resources to all ISA PnP devices; Auto assign ports to onboard COM ports if the corresponding item in Setup is set to "AUTO"
6Eh	Reserved
6Fh	Initialize floppy controller; Setup floppy related fields in 40:hardware
70h	Reserved
71h	Reserved
72h	Reserved
73h	Enter AWDFLASH.EXE if: AWDFLASH.EXE is found in floppy dive and ALT+F2 is pressed
74h	Reserved
75h	Detect and install all IDE devices: HDD, LS120, ZIP, CDROM...
76h	Reserved
77h	Detect serial ports and parallel ports
78h	Reserved
79h	Reserved

7Ah	Detect and install coprocessor
7Bh	Reserved
7Ch	Reserved
7Dh	Reserved
7Eh	Reserved
7Fh	Switch back to text mode if full screen logo is supported: if errors occur, report errors & wait for keys, if no errors occur or F1 key is pressed continue - Clear EPA or customization logo
80h	Reserved
81h	Reserved
82H	Call chipset power management hook: Recover the text fond used by EPA logo (not for full screen logo), If password is set, ask for password
83H	Save all data in stack back to CMOS
84h	Initialize ISA PnP boot devices
85h	Final USB initialization; NET PC: Build SYSID structure; Switch screen back to text mode; Set up ACPI table at top of memory; Invoke ISA adapter ROM's; Assign IRQ's to PCI devices; Initialize APM; Clear noise of IRQ's
86h	Reserved
87h	Reserved
88h	Reserved
89h	Reserved
90h	Reserved
91h	Reserved
92h	Reserved
93h	Read HDD boot sector information for Trend Anti-Virus code
94h	Enable L2 cache; Program boot up speed; Chipset final initialization; Power management final initialization; Clear screen and display summary table; Program K^ write allocation; Program P6 class write combining
95h	Program daylight saving; Update keyboard LED and typematic rate

96h	Build MP table; Build and update ESCD; Set CMOS century to 20h or 19h; Load CMOS time into DOS timer tick; Build MSIRQ routing table
FFh	Boot attempt (INT 19h)

AMI Win BIOS:

00	Control to Int 19 boot loader
01	Disable NMI
02	Power-on delay
03	Soft reset power-on
05	Disable cache
06	Uncompressed POST code
08	CMOS checksum
08	CMOS initialization
0A	CMOS initialization for date and time
0B	Initialization before keyboard batch
0C	Batch command to keyboard controller
0D	Verify batch command
0E	Initialize after KB controller batch
0F	Write KB command byte
10	Pin 23/24 block/unblock command
11	Check for <INS> key command
12	DMA/PIC disable
13	Chipset initialization
14	8254 timer test
19	Memory refresh test
20	Base 64K memory test
23	Set BIOS stack, setup before int. vector init
24	Interrupt vector initialization
25	Read input port of 9042 chip, clear password
26	Initialize global data for turbo switch
27	Initialize before setting video mode
28	Set video mode
2A	Initialize BUS
2B	Setup before operational video check

2C	Control to optional video ROM
2D	Proc. after optional video ROM routine
2E	Display memory Read/Write test if no EGA/VGA
2F	Display memory Read/Write test
30	Retrace check
31	Display alternate memory Read/Write check
32	Alternate display retrace check
34	Set display mode
37	Display power-on message
38	Initialize BUS types
39	Display BUS initialization error messages
3A	Display the hit <DEL> message
3B	Virtual modem memory test
40	Prepare descriptor tables
42	Enter virtual mode for memory test
43	Enable Interrupts for diagnostic mode
44	Initialize data to check memory wrap at 0:0
45	Check memory wrap, find total memory amount
46	Memory write test
47	640K base memory write test
48	Determine memory below 1MB
49	Determine memory above 1MB
4B	Check for soft reset, clear memory below 1MB
4C	Clear memory above 1MB
4D	Save memory size
4E	Display first 64K memory size
4F	Sequential and random memory test
50	Displayed memory size
51	Above 1MB memory test
52	Save memory size information
53	Enter real mode
54	Disable gate A-20 line

57	Adjust memory size
58	Clear hit <DEL> message
59	DMA/PIC test
60	DMA #1 base register test
62	DMA #2 base register test
65	Program DMA unit 1 and 2
66	Initialize 8259 Interrupt controller
67	Keyboard test
7F	Enable extended NMI sources
80	Stuck key and batch test
81	Keyboard controller test
82	Write command byte, initialize circular buffer
83	Lock key check
84	Compare memory size with CMOS
85	Password/soft error check
86	Programming before check
87	Execute CMOS setup
88	Programming after setup
89	Power-on display
8B	Shadow main and video BIOS
8C	Setup options after CMOS setup
8D	Initialize mouse
8E	Reset hard disk controller
8F	Floppy setup
91	Hard disk setup
94	Base/extended memory size
95	Init. PCI/VLB BUS optional ROM's from C800
96	Initialize before C800 optional ROM control
97	Control to optional ROM
98	Processing after optional ROM control
99	Setup timer data area/printer base address
9A	Set RS-232 base address

9B	Initialize before NPU test
9C	NPU initialization
9D	Initialization after NPU test
9E	Check extended KB, KB ID and num-lock
9F	Issue keyboard ID command
A0	Reset keyboard ID flag
A1	Cache memory test
A2	Display and soft errors
A4	Program memory wait states
A5	Clear screen, enable parity NMI
A7	Init. needed before control to E000 ROM
A8	Control to E000 ROM
A9	Init. needed after control to E000 ROM
AA	Display system configuration
B0	Uncompressed SETUP code for hot-key
B1	Copy any code to specific area
C2	Disable NMI, power-on delay
C5	Enable ROM, disable cache
C6	ROM BIOS checksum
C7	CMOS shutdown register test
C8	CMOS shutdown
CA	Initialize CMOS date and time
CB	Initialization before keyboard batch
CD	BAT command to keyboard controller
CE	Installation after keyboard controller batch
CF	Write keyboard command byte
D1	Check for <INS> key command
D2	Disable DMA and Interrupt controllers
D3	Chipset initialization/auto detect memory
D4	Uncompressed RUNTIME code
D5	RUNTIME code uncompressed
DD	Control to shadow RAM at F000:F000

AMI Ez-Flex BIOS:

01	NMI disabled; Start CPU flag test
02	Power on delay
03	Initialize system chipset
04	Check keyboard for soft/hard reset
05	Enable ROM
06	ROM BIOS checksum tested
07	8042 keyboard controller tested
08	8042 keyboard controller tested
09	8042 keyboard controller tested
0A	8042 keyboard controller tested
0B	8042 protected mode tested
0C	8042 keyboard controller tested
0D	CMOS RAM shutdown register tested
0E	CMOS checksum tested
0F	CMOS initialization
10	CMOS/RTC status OK
11	Disable DMA and PIC
12	Video display disabled
13	Chipset and memory initialized
14	8254 PIT tested
15	PIT channel 2 tested
16	PIT channel 1 tested
17	PIT channel 0 tested
18	PIT memory refresh tested
19	PIT memory refresh tested
1A	Check 15 microsecond refresh (PIT)
1B	Base 64K memory tested
20	Address lines tested
21	Base 64K parity memory tested

22	Memory Read/Write tested
23	Perform setup's prior to initialization of the vector table
24	Initialize BIOS vector table in lower 1KB of system RAM
25	8042 keyboard controller tested
26	Global for keyboard controller tested
27	Perform setups for vector table initialization
28	Monochrome video mode tested
29	Video (CGA) color mode tested
2A	Parity enable tested
2B	Check for optional ROM's
2C	Check for video ROM
2D	Determine if EGA/VGA is installed
2E	Video memory is tested if non EGA/VGA
2F	Video memory tested
30	Video adapter tested
31	Alternate video memory tested
32	Alternate video adapter tested
33	Video mode tested
34	Video mode tested
35	BIOS ROM data area initialized
36	Power on display cursor set
37	Power on message displayed
38	Cursor position read
39	Display cursor reference
3A	Display Setup message
40	Protected mode tested
41	Build descriptor tables
42	CPU enters protected mode
43	Protected mode interrupt enabled
44	Descriptor tables checked
45	Memory size checked
46	Memory read/Write tested

47	Base 640K memory tested
48	Memory below 1MB checked for
49	Memory above 1MB checked for
4A	ROM BIOS data area checked
4B	Memory below 1MB cleared for soft reset
4C	Memory above 1MB cleared for soft reset
4D	Update CMOS memory size
4E	Display base 64K memory test
4F	Memory test on base 640K performed
50	RAM size updated for shadow operation
51	Extended memory test performed
52	System is prepared for real mode
53	CPU is returned to real mode
54	CPU registers are returned to real mode
55	A20 gate disabled
56	BIOS data area rechecked
57	BIOS data area check complete
58	Setup message displayed
59	DMA register page tested
60	Display memory verified
61	DMA #1 tested
62	DMA #2 tested
63	Perform BIOS data area check
64	BIOS data area checked
65	DMA initialized
66	8259 PIC initialized
67	Keyboard tested
80	Keyboard reset
81	Check for stuck key and batch test
82	8042 keyboard controller tested
83	Lock key checked
84	Memory size compared to CMOS

85	Password and soft error checked
86	CMOS equipment checked performed
87	CMOS setup performed if selected
88	Main chipset reinitialized after CMOS setup
89	Power on message displayed
8A	Mouse check and wait message displayed
8B	Any ROM's attempted to be shadowed
8C	System initialized through CMOS settings
8D	Hard drives and floppy drives reset
8E	Floppy disk setup compared to CMOS settings
8F	Floppy controller initialized
90	Hard disks setup compared to CMOS settings
91	Hard disk controller initialized
92	BIOS data table checked
93	BIOS data table check complete
94	Memory size set
95	Display memory verified
96	All Interrupts cleared
97	Optional ROM's checked for
98	All Interrupts cleared
99	Timer data setup
9A	Serial ports checked for
9B	All Interrupts cleared
9C	Math coprocessor checked
9D	All Interrupts cleared
9E	Extended keyboard checked
9F	NumLock set on keyboard
A0	Keyboard reset
A1	Cache memory size tested
A2	Display any soft errors
A3	Typematic rate set
A4	Memory wait states set

A5	Display is cleared
A6	Parity and NMI enabled
A7	All Interrupts cleared
A8	System control is turned over to ROM at E0000
A9	All Interrupts cleared
AA	Displayed configuration
00	Call to Interrupt 19 for boot loader

Phoenix 4.0 BIOS:

02	Verify real mode
04	Get CPU type
06	Initialize system hardware
08	Initialize chipset registers with initial POST values
09	Set in POST flag
0A	Initialize CPU registers
0C	Initialize cache to initial POST values
0E	Initialize I/O
10	Initialize power management
11	Load alternate registers with initial POST values
12	Jump to UserPatch0
14	Initialize keyboard controller
16	BIOS ROM checksum
18	8254 programmable interrupt timer initialization
1A	8237 DMA controller initialization
1C	Reset 8254 programmable interrupt timer
20	Test DRAM refresh
22	Test 8742 keyboard controller
24	Set ES segment register to 4GB
28	Auto size DRAM
2A	Clear 512K base RAM
2C	Test 512K base address lines
2E	Test 512K base memory
32	Test CPU bus-clock frequency
37	Reinitialize the chipset
38	Shadow system BIOS ROM
39	Reinitialize the cache
3A	Auto size cache
3C	Configure advanced chipset registers

3D	Load alternate registers with CMOS values
40	Set initial CPU speed
42	Initialize interrupt vectors
44	Initialize BIOS interrupts
46	Check ROM copyright notice
48	Check video configuration against CMOS
49	Initialize PCI bus and devices
4A	Initialize all video adapters in system
4C	Shadow video BIOS ROM
4E	Display copyright notice
50	Display CPU type and speed
52	Test keyboard
54	Set key click if enabled
56	Enable keyboard
58	Test for unexpected interrupts
5A	Display prompt "Press F2 to Enter Setup"
5C	Test RAM between 512K and 640K
60	Test expanded memory
62	Test extended memory address lines
64	Jump to UserPatch1
66	Configure advanced cache registers
68	Enable external and CPU caches
6A	Display external cache size
6C	Display shadow message
6E	Display non-disposable segments
70	Display error messages
72	Check for configuration errors
74	Test real time clock
76	Check for keyboard errors
7C	Setup hardware interrupts vectors
7E	Test coprocessor if present
80	Disable onboard I/O ports

82	Detect and install external RS232 ports
84	Detect and install external parallel ports
86	Re-initialize on-board I/O ports
88	Initialize BIOS data area
8A	Initialize extended BIOS data area
8C	Initialize floppy controller
90	Initialize hard disk controller
91	Initialize local bus hard disk controller
92	Jump to UserPatch2
94	Disable A20 address line
96	Clear huge ES segment register
98	Search for option ROM's
9A	Shadow option ROM's
9C	Setup power management
9E	Enable hardware interrupts
A0	Set time of day
A2	Check key lock
A8	Erase F2 prompt
AA	Scan for F2 keystroke
AC	Enter setup
AE	Clear in-POST flag
B0	Check for errors
B2	POST done; prepare to boot operating system
B4	One beep
B6	Check password (optional)
B8	clear global descriptor table
BC	Clear parity checkers
BE	Clear screen (optional)
BF	Check virus and backup reminders
C0	Try to boot interrupt 19
D0	Interrupt handler error
D2	Unknown interrupt error

D4	Pending interrupt error
D6	Initialize option ROM error
D8	Shutdown error
DA	Extended block move
DC	Shutdown 10 error
E2	Initialize the chipset
E3	Initialize refresh counter
E4	Check for forced flash
E5	Check HW status of ROM
E6	BIOS ROM is ok
E7	Do a complete RAM test
E8	Do OEM initialization
E9	Initialize interrupt controller
EA	Read in bootstrap code
EB	Initialize all vectors
EC	Boot the flash program
ED	Initialize the boot device
EE	Boot code was read ok

Phoenix 4.0 Release 6.0:

02	Verify real mode
03	Disable non-maskable interrupt (NMI)
04	Get CPU type
06	Initialize system hardware
07	Disable shadow and execute code from the ROM
08	Initialize chipset with initial POST values
09	Set IN POST flag
0A	Initialize CPU registers
0B	Enable CPU cache
0C	Initialize caches to initial POST values
0E	Initialize I/O component
0F	Initialize the local bus IDE
10	initialize power management
11	Load alternate registers with initial POST values
12	Restore CPU control word during warm boot
13	Initialize PCI bus mastering devices
14	Initialize keyboard controller
16	BIOS ROM checksum
17	Initialize cache before memory auto size
18	8254 programmable interrupt timer initialization
1A	8237 DMA controller initialization
1C	Reset programmable interrupt controller
20	Test DRAM refresh
22	Test 8742 keyboard controller
24	Set ES segment register to 4GB
26	Enable gate A20 line
28	Auto size DRAM
29	Initialize POST memory manager
2A	Clear 512KB base RAM

2C	RAM failure on address line *xxxx*
2E	RAM failure on data bits *xxxx* of low byte of memory bus
2F	Enable cache before system BIOS shadow
30	RAM failure on data bits *xxxx* of high byte of memory bus
32	Test CPU bus clock frequency
33	Initialize Phoenix Dispatch Manager
36	Warm start shut down
38	Shadow system BIOS ROM
3A	Auto size cache
3C	Advanced configuration of chipset registers
3D	Load alternate registers with CMOS values
41	Initialize extended memory for RomPilot
42	Initialize interrupt vectors
45	POST device initialization
46	Check ROM copyright notice
47	Initialize I20 support
48	Check video configuration against CMOS
49	Initialize PCI bus and devices
4A	Initialize all video adapters in system
4B	Quiet Boot start (optional)
4C	Shadow video BIOS ROM
4E	Display BIOS copyright notice
4F	Initialize MultiBoot
50	Display CPU type and speed
51	Initialize EISA board
52	Test keyboard
54	Set key click if enabled
55	Enable USB devices
58	Test for unexpected interrupts
59	Initialize POST display service
5A	Display prompt "Press F2 to enter SETUP"
5B	Disable CPU cache

5C	Test RAM between 512KB and 640KB
60	Test extended memory
62	Test extended memory address lines
64	Jump to UserPatch1
66	Configure advanced cache registers
67	Initialize Multi Processor APIC
68	Enable external and CPU caches
69	Setup system management mode (SMM) area
6A	Display external L2 cache size
6B	Load custom defaults (optional)
6C	Display shadow area message
6E	Display possible high address for UMB recovery
70	Display error messages
72	Check for configuration errors
76	Check for keyboard errors
7C	Set up hardware interrupt vectors
7D	Initialize Intelligent System Monitoring
7E	Initialize coprocessor if present
80	Disable onboard super I/O ports and IRQ's
81	Late POST device initialization
82	Detect and install external RS232 ports
83	Configure non-MCD IDE controllers
84	Detect and install external parallel ports
85	Initialize PC compatible PnP ISA devices
86	Reinitialize onboard I/O ports
87	Configure motherboard configurable devices (optional)
88	Initialize BIOS data area
89	Enable non-maskable interrupts (NMI's)
8A	Initialize extended BIOS data area
8B	Test and initialize PS/2 mouse
8C	Initialize floppy controller
8F	Determine number of ATA drives (optional)

90	Initialize hard disk controllers
91	Initialize local bus hard disk controllers
92	Jump to UserPatch2
93	Build MPTABLE for multi processor boards
95	Install CD ROM for boot
96	Clear huge ES segment register
97	Fix up multi processor table
98	Search for option ROM's
99	Check for SMART drive (optional)
9A	Shadow option ROM's
9C	Set up power management
9D	Initialize security engine (optional)
9E	Enable hardware interrupts
9F	Determine number of ATA and SCSI drives
A0	Set time of day
A2	Check key lock
A4	Initialize typematic rate
A8	Erase F2 prompt
AA	Scan for F2 key stroke
AC	Enter setup
AE	Clear boot flag
B0	Check for errors
B1	Inform Rom Pilot about the end of POST
B2	POST done - prepare to boot operating system
B4	One short beep
B5	Terminate Quiet Boot (optional)
B6	Check password
B7	Initialize ACPI BIOS
B9	Prepare boot
BA	Initialize DMI parameters
BB	Initialize PnP option ROM's
BC	Clear parity checkers

BD	Display multiboot menu
BE	Clear screen
BF	Check virus and backup reminders
C0	Try to boot with interrupt 19
C1	Initialize POST Error Manager (PEM)
C2	Initialize error logging
C3	Initialize error display function
C4	Initialize system error handler
C5	PnP dual CMOS (optional)
C6	Initialize notebook docking (optional)
C7	Initialize notebook docking late
C8	Force check (optional)
C9	Extended checksum (optional)
CA	Redirect Int 15h to enable remote keyboard
CB	Redirect Int 13 to Memory Technologies Devices such as ROM, RAM, PCMCIA, and serial disk
CC	Redirect Int 10h to enable remote serial video
CD	Re-map I/O and memory for PCMCIA
CE	Initialize digitizer and display message
D2	Unknown interrupt
	The following are for boot block in Flash ROM
E0	Initialize the chipset
E1	Initialize the bridge
E2	Initialize the CPU
E3	Initialize the system timer
E4	Initialize system I/O
E5	Check force recovery boot
E6	Checksum BIOS ROM
E7	Go to BIOS
E8	Set Huge Segment
E9	Initialize Multi Processor
EA	Initialize OEM special code

EB	initialize PIC and DMA
EC	Initialize Memory type
ED	Initialize Memory size
EE	Shadow Boot Block
EF	System memory test
F0	Initialize interrupt vectors
F1	Initialize Run Time Clock
F2	Initialize video
F3	Initialize System Management Manager
F4	Output one beep
F5	Clear Huge Segment
F6	Boot to mini DOS
F7	Boot to Full DOS

Award BIOS Beep Codes:

Beeps	Error Message	Description
1long, 2 short	Video adapter error	Either video adapter is bad or is not seated properly. Also, check to ensure the monitor cable is connected properly.
Repeating (endless loop)	Memory error	Check for improperly seated or missing memory.
1long, 3short	No video card or bad video RAM	Reseat or replace the video card.
High frequency beeps while running	Overheated CPU	Check the CPU fan for proper operation. Check the case for proper air flow.
Repeating High/Low	CPU	Either the CPU is not seated properly or the CPU is damaged. May also be due to excess heat. Check the CPU fan or BIOS settings for proper fan speed.

AMI BIOS Beep Codes:

Beeps	Error Message	Description
1 short	DRAM refresh failure	The programmable interrupt timer or programmable interrupt controller has probably failed
2 short	Memory parity error	A memory parity error has occurred in the first 64K of RAM. The RAM IC is probably bad
3 short	Base 64K memory failure	A memory failure has occurred in the first 64K of RAM. The RAM IC is probably bad
4 short	System timer failure	The system clock/timer IC has failed or there is a memory error in the first bank of memory
5 short	Processor error	The system CPU has failed
6 short	Gate A20 failure	The keyboard controller IC has failed, which is not allowing Gate A20 to switch the processor to protected mode. Replace the keyboard controller
7 short	Virtual mode processor exception error	The CPU has generated an exception error because of a fault in the CPU or motherboard circuitry
8 short	Display memory read/write error	The system video adapter is missing or defective
9 short	ROM checksum error	The content of the system BIOS ROM does not match the expected checksum value. The BIOS ROM is probably defective and should be replaced
10 short	CMOS shutdown register read/write error	The shutdown for the CMOS has failed
11 short	Cache error	The L2 cache is faulty

1 long, 2 short	Failure in video system	An error was encountered in the video BIOS ROM, or a horizontal retrace failure has been encountered
1 long, 3 short	Memory test failure	A fault has been detected in memory above 64KB
1 long, 8 short	Display test failure	The video adapter is either missing or defective
2 short	POST Failure	One of the hardware tested have failed
1 long	POST has passed all tests	